

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК 004.021

«До захисту допущено»
Завідувач кафедри

_____ О.В. Коваль
(підпис) (ініціали, прізвище)

“___” _____ 2018р.

Магістерська дисертація

зі спеціальності - 121 Інженерія програмного забезпечення
за спеціалізацією - Програмне забезпечення веб-технологій та мобільних
пристроїв
на тему: Обробка гідроакустичних сигналів за допомогою
вейвлетів

Виконав (-ла): студент (-ка) 6 курсу, групи ТІ-71мп
Шевчук Олександр Олегович
(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник доц., к.т.н. Варава Іван Андрійович
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Рецензент _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ - 2018

**Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 121 Інженерія програмного забезпечення

за спеціалізацією - Програмне забезпечення веб-технологій та мобільних пристроїв

ЗАТВЕРДЖУЮ

Завідувач кафедри

Коваль О.В.

(прізвище, ініціали)

(підпис)

«_____» _____ 2018р.

**З А В Д А Н Н Я
НА МАГІСТЕРСКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Шевчук Олександр Олегович

(прізвище, ім'я, по батькові)

1. Тема дисертації Обробка гідроакустичних сигналів за допомогою вейвлетів

Науковий керівник к.т.н., доцент Варава І.А.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “___” _____ 20__ року №___

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження алгоритми фільтрації сигналів та засоби розробки програмного забезпечення для реалізації обраного алгоритму

4. Предмет дослідження алгоритм фільтрації гідроакустичних сигналів за допомогою дискретного вейвлет-перетворення, пакет прикладних програм для реалізації десктопного застосунку

5. Перелік питань, які потрібно розробити _____

1) Вивчити процес фільтрації сигналів _____

2) Вивчити процес дискретного вейвлет-перетворення _____

3) Ознайомитися з алгоритмом фільтрації гідроакустичних сигналів за допомогою вейвлетів _____

4) Розробити програмне забезпечення для реалізації алгоритму фільтрації гідроакустичних сигналів за допомогою дискретного вейвлет-перетворення _____

6. Орієнтований перелік ілюстративного матеріалу _____
_____мета, постановка задачі, алгоритм фільтрації гідроакустичного сигналу, архітектура
системи, результати дослідження, графічне відображення відфільтрованого сигналу

7. Орієнтований перелік публікацій _____
Обробка гідроакустичних сигналів за допомогою вейвлетів // XVI міжнародна
науково-практична конференція аспірантів, магістрантів, студентів «Сучасні проблеми
наукового забезпечення енергетики», м. Київ, 24-27 квітня 2018р. У К.: КПІ ім. Ігоря
Сікорського», 2018. – С. 243. _____

8. Дата видачі завдання «_____»_____201____р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Затвердження теми роботи	17.05.2018	
2	Вивчення та аналіз роботи	01.05.2018-03.09.2018	
3	Розробка архітектури та загальної структури системи	03.09.2018-28.09.2018	
4	Розробка структури окремих підсистем	01.10.2018-26.10.2018	
5	Програмна реалізація системи	22.10.2018	
6	Оформлення пояснювальної записки	02.09.2018-10.12.2018	
7	Захист програмного продукту	20.11.2018-26.11.2018	
8	Передзахист	26.11.2018	
9	Захист	19.12.2018	

Студент

Науковий керівник

(підпис)

(підпис)

(прізвище та ініціали)
Шевчук О.О.

(прізвище та ініціали)
Варава І.А.

РЕФЕРАТ

Структура й обсяг дипломної роботи

Магістерська дисертація складається зі вступу, семи розділів, вступу, висновку, переліку посилань з 30 найменувань, 1 додатку і містить 26 рисунків, 21 таблицю. Повний обсяг магістерської дисертації складає 80 сторінок, з яких перелік посилань займає 3 сторінки.

Актуальність теми. На сьогоднішній день, гідроакустика стикається з проблемою розробки теорій, які описують поширення звуку в морській воді, а також створення адекватних математичних моделей, що пояснюють експериментальні сигнали. Можливим вирішенням є створення теорій на базі вейвлетів. Наразі, одним із основних інструментів обробки гідроакустичних сигналів є перетворення Фур'є. Проте воно дає інформацію тільки про частоту, яка присутня в сигналі, але не дає ніякої інформації про те, протягом якого проміжку часу ця частота присутня. А використання вейвлет-аналізу може частково вирішити цю проблему за допомогою використання різних конфігурацій вейвлетів, які реалізують різні варіанти співвідношення невизначеності. Також важливою є проблема фільтрації отриманих даних. Для вирішення цих проблем було вирішено створити зручне, інтуїтивно зрозуміле десктопне застосування, за допомогою якого можна провести вейвлет-аналіз та фільтрацію гідроакустичних сигналів.

Зв'язок роботи з науковими програмами, планами, темами

Дисертаційна робота магістра виконувалась у КПІ ім. Ігоря Сікорського у відповідності з планом наукових досліджень кафедри АПЕПС.

Мета дослідження. Метою роботи є удосконалення процесу фільтрації гідроакустичних сигналів та розробка десктопного застосунку, що буде проводити розклад сигналу за допомогою дискретного вейвлет-перетворення, його фільтрації та представлення результатів в зручному для користувача вигляді.

Для реалізації поставленої мети були сформульовані наступні **завдання**

дослідження, що визначили логіку дослідження та його структуру:

- на основі аналізу методів обробки сигналів обґрунтувати використання дискретного вейвлет-перетворення;
- розробити структуру класів програмного забезпечення;
- реалізувати застосунок для фільтрації гідроакустичних сигналів за допомогою вейвлетів.

Об'єктом дослідження є програмне забезпечення для обробки гідроакустичних сигналів.

Предмет дослідження є програмне забезпечення для фільтрації гідроакустичних сигналів.

Методи дослідження: При вирішенні задач роботи застосовувались наступні методи:

- аналіз гідроакустичних сигналів за допомогою перетворення Фур'є;
- аналіз гідроакустичних сигналів за допомогою вейвлетів.

Наукова новизна одержаних результатів. Найбільш суттєвими науковими результатами магістерської дисертації є набуття подальшого розвитку методів фільтрації гідроакустичних сигналів.

Практичне значення одержаних результатів роботи полягає в розробці десктопного застосунку для фільтрації сигналу за допомогою дискретного вейвлет-перетворення, що може використовуватись для аналізу гідроакустичних сигналів.

Ключові слова. *ВЕЙВЛЕТ, МАТЕРИНСЬКИЙ ВЕЙВЛЕТ, ВЕЙВЛЕТ-ПЕРЕТВОРЕННЯ, ГІДРОАКУСТИКА.*

ABSTRACT

The structure and volume of the thesis.

Master's thesis consists of an introduction, seven chapters, conclusion, list of references with 30 titles, 1 annex, and contains 25 figures, 21 tables. The full range of master's thesis is 80 pages with a list of links takes 3 pages.

Actuality of theme. In recent years, hydroacoustics faces the problem of developing theories describing the propagation of sound in seawater, as well as the creation of adequate mathematical models that explain the experimental signals. A possible solution is to create theories based on wavelets. Currently, one of the main tools for processing hydroacoustic signals is the Fourier transform. However, it only provides information about the frequency that is present in the signal, but does not give any information about how much time this frequency is present. And the use of wavelet analysis can partially solve this problem by using different wavelet configurations that implement different variants of the uncertainty relation. Also important problem is filtration of the received data. To solve these problems it was decided to create a convenient, intuitive desktop application, with which you can perform wavelet analysis and filtering of hydroacoustic signals.

Relationship with academic programs, plans, themes.

Master's thesis performed at NTUU "KPI" in accordance with the plan of research APEPS.

The aim of the study. The aim of the work is to improve the process of filtration of hydroacoustic signals and develop a desktop software that will perform discrete wavelet transform for signal, filter it and present the results in user-friendly form.

To achieve this goal the following objectives were formulated research **following tasks** the logic and structure:

- based on the analysis of signal processing methods to justify using discrete wavelet transformation;
- develop the structure of the software classes;

- implement an application for filtering hydroacoustic signals using wavelets.

Object is software for processing hydroacoustic signals.

The subject is software for the filtration of hydroacoustic signals.

Methods: In solving problems of applied following methods:

- analysis of hydroacoustic signals using the Fourier transform;
- analysis of hydroacoustic signals using wavelets.

Scientific novelty of the results. The most significant scientific results of the master thesis is the adaptation of methods of filtration of hydroacoustic signals.

The practical significance of the results is to develop a desktop application for signal filtering using a discrete wavelet transform, which can be used for the analysis of hydroacoustic signals.

Keywords. *WAVELET, MATERIAN WAVELET, WAVELET-TRANSFORMATION, HYDROACOUSTICS.*

ЗМІСТ

Перелік умовних скорочень і позначень	9
Вступ.....	10
1 Опис предметної області.....	11
1.1 Проблема фільтрації звукових сигналів	11
1.2 Класифікація типів і джерел шумів	12
1.3 Класифікація видалення шумів	13
2 Існуючі рішення для аналізу та фільтрації звукових даних.....	14
2.1 Wavelet toolbox для matlab.....	14
2.2 Matcad wavelets extension pack.....	16
2.3 Wavelet explorer	17
3 Особливості аналізу сигналів.....	19
3.1 Перетворення Фур'є.....	19
3.2 Недоліки перетворення Фур'є.....	20
3.3 Вейвлет-перетворення.....	21
3.4 Неперервне вейвлет-перетворення.....	22
3.5 Дискретне вейвлет-перетворення.....	25
3.6 Зворотне дискретне вейвлет-перетворення.....	28
3.7 Фільтрація сигналу та обчислення порогового значення	28
4 Технологічна база розробленого програмного засобу.....	32
4.1 Основні відомості про мову програмування C#.....	32
4.2 Концепція .Net	33
4.3 Архітектура .Net Framework	34
4.4 Windows Presentation Foundation	36
4.5 Model-View-ViewModel.....	38
4.6 XAML	40
4.7 Dependency injection.....	41
4.8 Середовище розробки Visual Studio	41
5 Реалізація системи	43

5.1 Структура проекту	43
5.2 Обробка вхідних даних	44
5.3 Реалізація модулю обчислення дискретного вейвлет-перетворення	45
5.4 Реалізація модулю обчислення швидкого перетворення Фур'є.....	47
5.5 Графічна інтерпретація даних	48
6 Приклад роботи системи	49
6.1 Інтерфейс програмного забезпечення.....	49
6.2 Завантаження сигналу до системи	50
6.3 Конфігурування параметрів обробки сигналу	51
7 Стартап.....	55
7.1 Опис ідеї проекту	55
7.2 Технологічний аудит ідеї проекту.....	57
7.3 Аналіз ринкових можливостей запуску стартап-проекту	59
7.4 Розробка ринкової стратегії проекту.....	66
7.5 Розроблення маркетингової програми стартап-проекту	69
Висновки.....	73
Перелік використаних джерел	74
Додаток А.....	77
Додаток Б	80

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

CWT - Continuous Wavelet Transform;

DFT - Discrete Fourier Transform;

DWT - Discrete Wavelet Transform;

FFT - Fast Fourier Transform;

IDWT - Inverse Discrete Wavelet Transform

STFT - Short-time Fourier Transform;

ВСТУП

Гідроакустика отримала широке практичне застосування – її використовують для підводної локації, зв'язку, океанологічних досліджень, вирішення військових задач. Звук дуже ефективно поширюється через воду, так що його можна чути і виявляти на великих відстанях.

Одним із способів вирішення поставлених перед гідроакустикою задач є використання вейвлетів. Вейвлет-перетворення є потужним інструментом у сфері обробки сигналів, стискання даних та зменшенні шуму. Більше того, вони починають підривати основну математичну техніку в аналізі сигналів: перетворення Фур'є. При цьому вони відкривають новий спосіб зрозуміти та дослідити сигнали. Перетворення Фур'є дає інформацію тільки про частоту, яка присутня в сигналі, і не дає ніякої інформації про те, в який проміжок часу ця частота присутня. Таким чином воно за своєю суттю не може відрізнити стаціонарний сигнал від нестаціонарного, що є великою проблемою для його застосовності [1].

Саме тому система, що розробляється, базується на вейвлет-перетворенні. Вона пропонує користувачу завантажити аудіо-файл гідроакустичного сигналу в форматі RAD та виконати його фільтрацію за допомогою дискретного вейвлет перетворення.

Відмінною особливістю вейвлет-аналізу є те, що в ньому можна використовувати сімейства функцій, що реалізують різні варіанти співвідношення невизначеності. Відповідно, користувач має можливість гнучкого вибору між ними і застосування тих вейвлетних функцій, які найбільш ефективно вирішують поставлені завдання [2]. Таким чином, враховуючи багатоканальність гідроакустичного сигналу та результати, які можна отримати після їх вейвлет-аналізу, можна отримати значних успіхів у вирішенні задач поставлених перед гідроакустикою – визначення напрямку руху і відстані до об'єкту.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Проблема фільтрації звукових сигналів

Загальна проблема, що стоїть перед аналізом даних, в контексті обробки сигналів, як відфільтрувати шум від основного сигналу. Крім цього, у виробництві існує багато інших проблемних питань даних, таких як:

- Питання збору даних.
- Відсутність даних.
- Екзогенні чинники, такі як автомасштабування або зміна вхідного трафіку.
- Зміни умовного розподілу вихідних даних заданих на вході, тоді як розподіл входу може залишатися незмінним.

Наведений нижче приклад (рис. 1.1) демонструє отриманий сигнал (синій) із шумом та основним сигналом без шуму (червоним кольором).

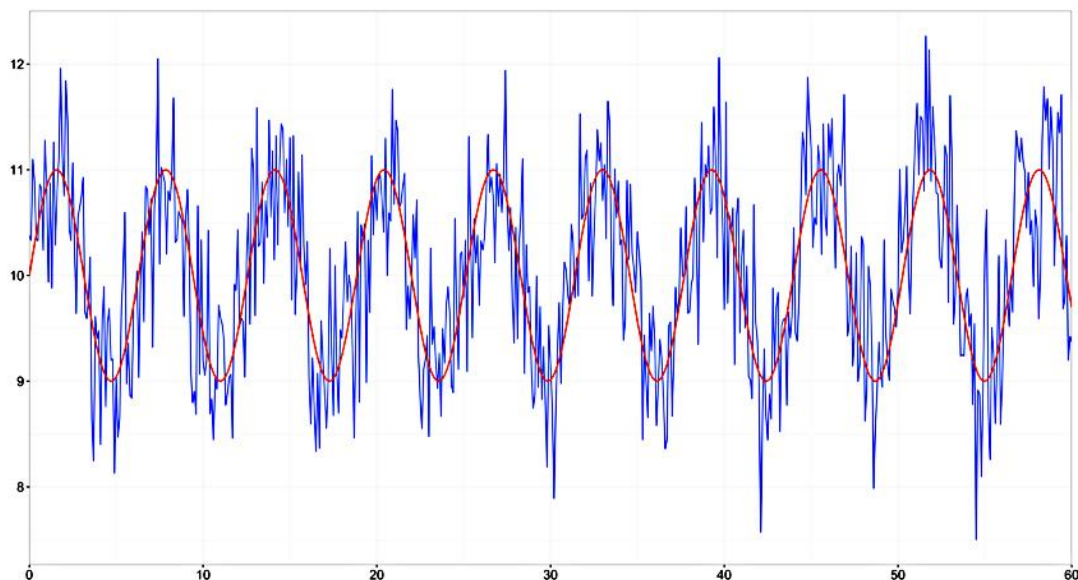


Рисунок 1.1 – Осцилограма зашумленого сигналу

На відміну від наведеного вище прикладу, який піддається візуальному аналізу, у більшості випадків фільтрація шуму для визначення сигналу не є можливим за допомогою візуального аналізу. Більш важливим є те, що з урахуванням обсягу

кількості часових рядів, проведення візуального аналізу не є практичним. Необхідно проводити алгоритмічний аналіз даних.[3]

Крім того, видалення шуму із сигналу не є самоціллю. У виробничому середовищі важливим є отримання усвідомленої статистики з сигналу.

1.2 Класифікація типів і джерел шумів

Є багато типів і джерел шуму або викривлення і вони включають:

- Електронні шуми, такі як тепловий шум і «стріляючий» шум.
- Акустичний шум, що виникає при русі, вібраційний або зіткнення джерел, таких як обертальні машини, рухомі транспортні засоби, натискання клавіатури, вітер та дощ.
- Електромагнітний шум, який може перешкоджати передачі та прийому аудіо сигналу.

Викривлення сигналу - це термін, який часто використовується для опису систематичної небажаної зміни сигналу та посиляється на зміни сигналу від неідеальних характеристик каналу зв'язку, реверберації затухання сигналу, еха та відсутніх семплів [4]. Залежно від частоти, спектру або часових характеристик, шум далі класифікується у декілька категорій:

- Білий шум: чисто випадковий шум має імпульсну автокореляційну функцію та плоский спектр потужності. Білий шум теоретично містить всі частоти в рівній потужності.
- Вузькосмуговий шум: це шумовий процес із вузькою смугою пропускання, наприклад, 50/60 Гц від джерела живлення.
- Кольоровий шум: це не білий шум або будь-який широкосмуговий шум, спектр якого має не плоску форму. Прикладами є рожевий шум, коричневий шум та авторегресійний шум.
- Імпульсний шум: складається з короткочасних імпульсів випадкової амплітуди, часу виникнення та тривалості.

- Тривалі шумові імпульси: складаються з відносно довготривалих шумових імпульсів, такі як клацання.

1.3 Класифікація видалення шумів

Видалення шумів із звуку можна розділити на два види - комплементарний тип і не комплементарний тип. Комплементарний тип включає в себе стиснення аудіосигналу і правильний спосіб запису. Не комплементарний тип є ефективною методикою для зменшення рівня шуму, який вже існує у вихідному матеріалі [5].

Як аналогові, так і цифрові пристрої мають особливу характеристику, що робить їх схильними до шуму. Існує активний шумовий контроль, який також називається шумозаглушенням або активним шумозаглушенням - це техніка для зменшення непотрібного звуку та звуку, шляхом додавання другого звуку, спеціально розробленого для скасування існуючого [6]. Звук - це аналоговий сигнал, який працює на частоті, яка складається з фази стискування та фази розрідження.

Одним із не комплементарних способів фільтрації є використання вейвлет-перетворення. Воно складається з безлічі основних функцій, які можуть бути використані для аналізу сигналів у одночасному режимі часу і частоти. Цей аналіз досягається використанням масштабованого вікна для покриття частотно-часової площини, що забезпечує зручний засіб для аналізу нестационарного сигналу, який часто зустрічається в більшості випадків. Вейвлет-аналіз приймає вейвлет-функцію прототипу, відомий як материнський вейвлет. Цей вейвлет у поворотах генерує набір базових функцій, відомих як дочірні вейвлети за допомогою рекурсивного масштабування та переміщення.

2 ІСНУЮЧІ РІШЕННЯ ДЛЯ АНАЛІЗУ ТА ФІЛЬТРАЦІЇ ЗВУКОВИХ ДАНИХ

2.1 Wavelet Toolbox для MatLab

Wavelet Toolbox надає функції та програми для аналізу та синтезу сигналів та зображень. Набір інструментів включає алгоритми для безперервного вейвлет-аналізу, вейвлет-когерентності, синхросигнування та адаптивного аналізу часу. Панель інструментів також включає додатки та функції для дискретного вейвлет-аналізу сигналів та зображень, включаючи вейвлет-пакети та перетворення з подвійним деревом [7].

Використовуючи безперервний аналіз вейвлет, ви можете дізнатись про те, як спектральні властивості розвиваються з часом, визначати загальні патерни, що змінюються за часом для двох сигналів, і виконувати фільтрацію, локалізовану за часом. Використовуючи дискретний вейвлет аналіз, ви можете аналізувати сигнали та зображення за різною роздільною здатністю для виявлення змінних точок, розривів та інших подій, які не можуть бути легко видимими у вихідних даних. Ви можете порівнювати статистику сигналів у кількох масштабах та виконувати фрактальний аналіз даних, щоб виявити приховані шаблони.

За допомогою Wavelet Toolbox можна отримати рідкісне представлення даних, корисних для видалення шуму або стиснення даних, зберігаючи важливі функції. Багато функцій підтримують генерацію коду C / C ++.

Перевагами даної системи є: широкий вибір інструментів та методик для обробки та аналізу сигналів.

Серед основних недоліків можна виділити те, що даний програмний продукт не є безкоштовним.

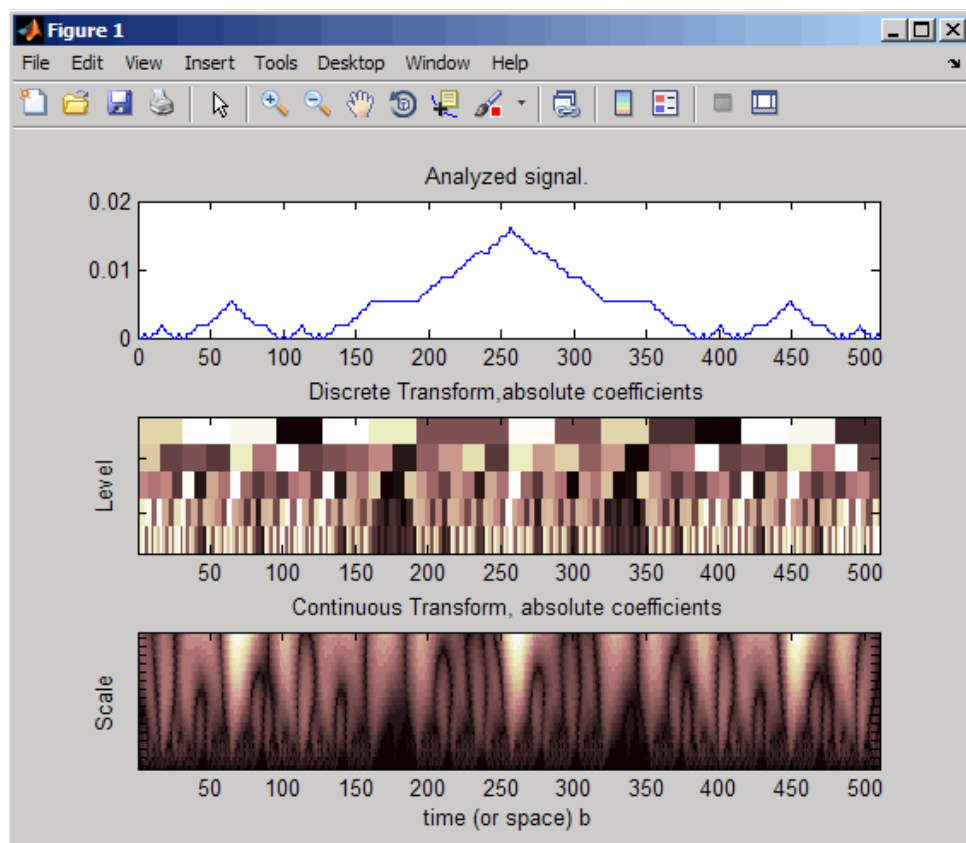


Рисунок 2.1 – Неперервне вейвлет перетворення у Wavelet Toolbox

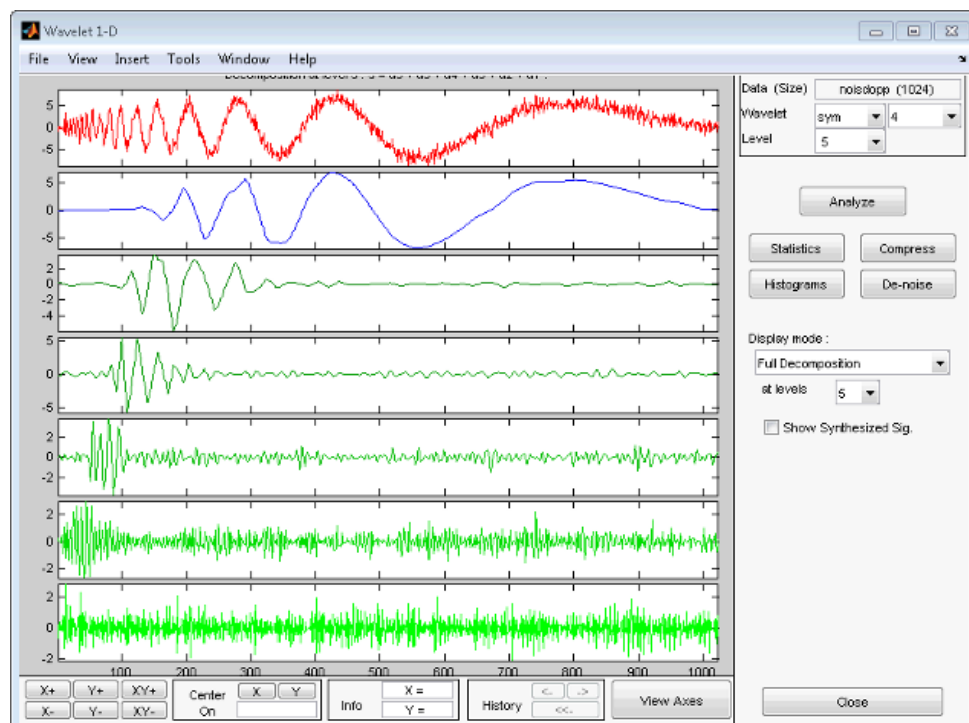


Рисунок 2.2 – Дискретне вейвлет перетворення у Wavelet Toolbox

2.2 Mathcad Wavelets Extension Pack

Пакет Wavelets Extension Pack дозволяє застосовувати новий підхід до аналізу сигналів та зображень, аналізу часових рядів, статистичної оцінки сигналів, аналізу стиснення даних та спеціальних числових методів. Створить майже безмежну кількість функцій, які дублюють будь-яке природне або абстрактне середовище. Функціональність включає в себе одно- та двомірні вейвлети, дискретне вейвлет-перетворення, аналіз з множинним дозволом тощо [8].

Wavelets Extension Pack підтримує базові та розширені версії програми. Утиліта Wavelets Extension Pack інтегрує понад 60 ключових функцій вейвлету, конкуруючи з аналогічними інструментами MATLAB і Mathematica за більш доступною ціною. Продукт включає ортогональні та біортогональні сімейства вейвлетів, включаючи вейвлети Хаара, Даубельта, Симмелти, Койфлети і Бі-сплайни.

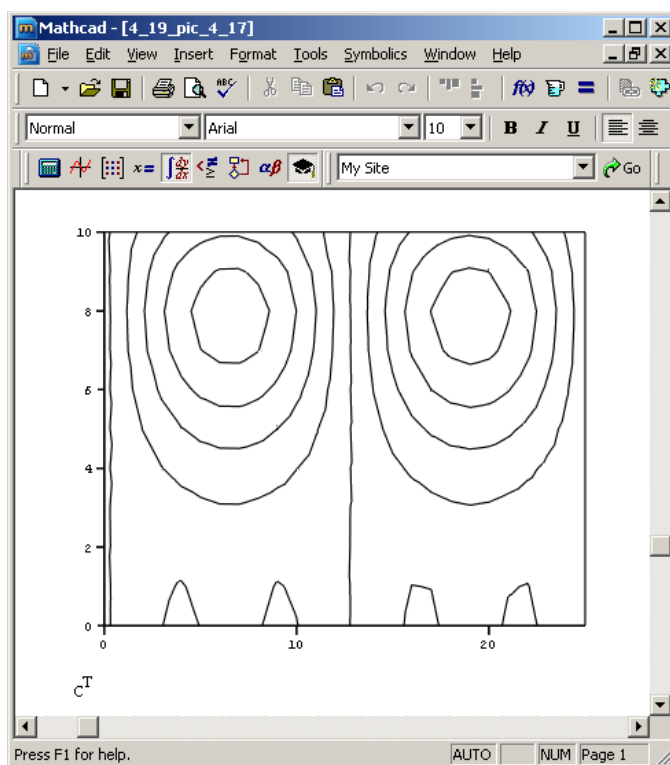


Рисунок 2.3 – Вейвлет-спектр сигналу на основі "мексиканського капелюха"

Популярний інтерфейс Mathcad забезпечує найкращу простоту використання та швидке застосування. Універсальне середовище Mathcad ідеально підходить для експериментів з вейвлетами та виконання сценаріїв if-if. Крім того, чудова інтеграція

Wavelets Extension Pack з Mathcad та іншими додатковими інструментами (включаючи процеси обробки сигналів та розширення зображень) дає вам більше можливостей, ніж будь-коли. Також присутня велика інтерактивна документація про основи вейвлетів, програми, приклади та довідкові таблиці.

2.3 Wavelet Explorer

Вбудовані функції і утиліти Wavelet Explorer дозволяють застосовувати різні вейвлет-перетворення в різних проектах. За допомогою пакету можна генерувати фільтри загального застосування, такі як фільтри екстремальної фази Добеши і фільтри мінімальної асиметрії, сплайнові фільтри та інші. Можна відображати на екрані окремі вейвлети і вейвлет-пакети та збільшено їх деталі. Можна перетворювати дані за допомогою великого числа вейвлет-базисів, базисів вейвлет-пакетів або локальних тригонометричних базисів і робити зворотні перетворення в одному і двох вимірах. Розглянути перетворення в частотно-часовому (фазовому) просторі, вибираючи різні базиси і початкові умови. Стиснення даних і видалення шуму стають доволі простими процедурами завдяки вбудованим функціям Wavelet Explorer [9].

Крім того, що Wavelet Explorer має велику колекцією потужних інструментів аналізу і візуалізації, цей пакет є відмінним інтерактивним посібником для початківців у вивченні вейвлет-теорії. Ясні приклади розкажуть про основи вейвлетів і як досліджувати їх властивості, потім продемонструють, як застосувати методи вейвлет-аналізу в конкретній галузі.

Написані на мові Mathematica, всі вбудовані функції і утиліти Wavelet Explorer повністю програмовані.

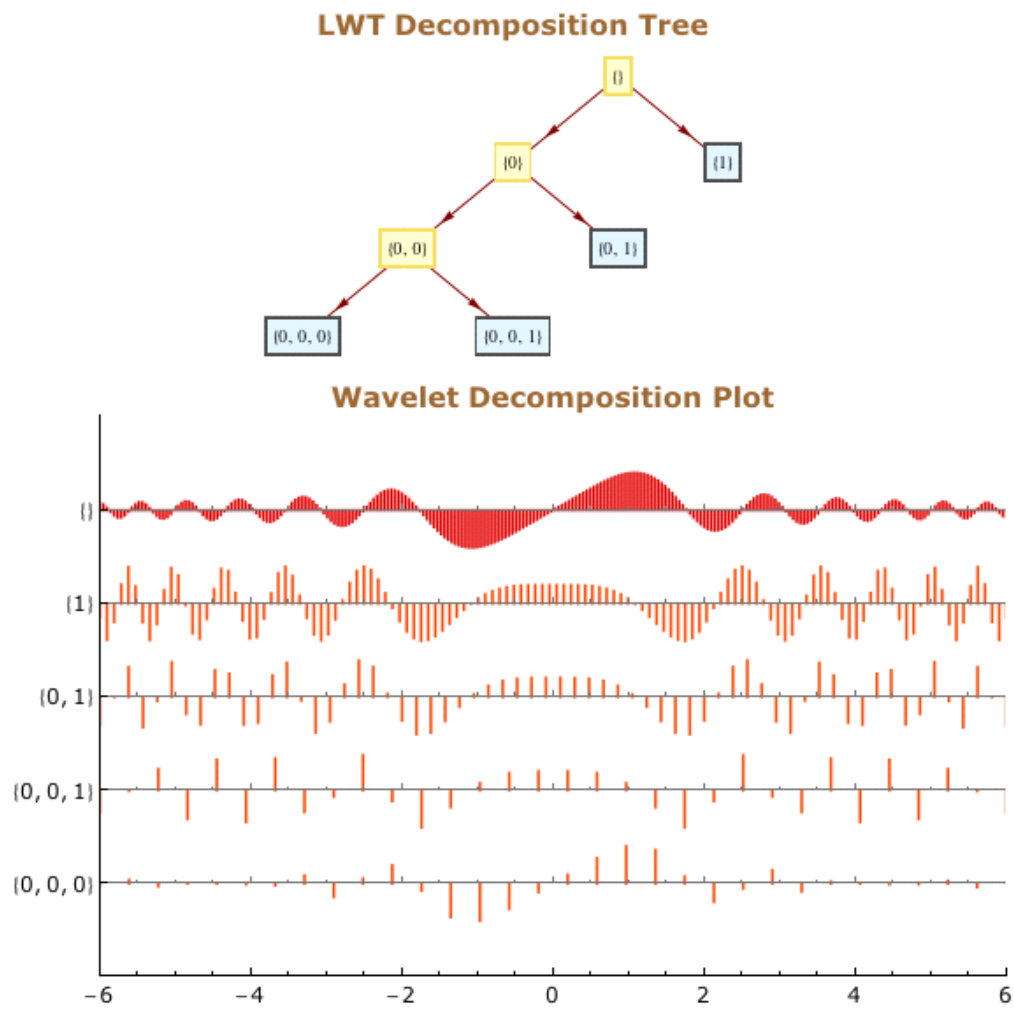


Рисунок 2.4 – Дискретне вейвлет-перетворення у Wavelet Explorer

3 ОСОБЛИВОСТІ АНАЛІЗУ СИГНАЛІВ

3.1 Перетворення Фур'є

Перетворення Фур'є було розроблено французьким математиком Жаном Батистом Д'Жозефом Фур'є у 1822 р. у книзі "Théorie analytique de la chaleur". Сигнал, отриманий імпульсним радаром, являє собою часову послідовність імпульсів, для яких вимірюється амплітуда і фаза. Методи обробки доплера базуються на вимірюванні спектрального (частотного) вмісту цього сигналу. Частотний зміст цього сигналу часової області отримується шляхом виконання його Фур'є перетворення, представляючи його як сигнал частотної області або спектр сигналу часової області.

Фур'є-перетворення стало основним методом обробки сигналів, оскільки радіолокаційне відлуння містить різноманітну інформацію в формі сигналу. Ця інформація конвертована перетворенням Фур'є у формат даних, який може використовуватися автоматизованою обробкою сигналів [10].

Перетворення Фур'є є формою аналізу, що бере за основу форму сигналу у вигляді синусів та косинусів (базові функції), а це означає, що форма сигналу розписана як сума синусних або косинусних хвиль різної частоти, фази та амплітуди. Цей аналіз називається дискретним перетворенням Фур'є (DFT).

На жаль, конвертація - це дуже трудомістка процедура. Однак швидке перетворення Фур'є (FFT) є програмним варіантом швидкого алгоритму. Швидкість аналізу досягається за допомогою:

- Вибору відповідного математичного алгоритму.
- Уникнення множення (множення вимагає більшого часу виконання, ніж додавання).
- Перші результати ще можуть бути доступними під час розрахунку.

За допомогою швидкого аналізу Фур'є ціла сигнальна форма радарного ехо може зберігатися як невеликий набір даних за допомогою цифрової обробки сигналів. Ці дані можуть використовуватися для ідентифікації радіолокаційних цілей.

3.2 Недоліки перетворення Фур'є

Великий недолік розширення Фур'є полягає в тому, що він має тільки частотну роздільну здатність і немає часової роздільної здатності. Це означає, що хоча ми і зможемо визначаємо всі частоти, присутні в сигналі, ми не знаємо, коли вони присутні. Щоб подолати цю проблему в останні десятиріччя було розроблено кілька рішень, які більш-менш здатні представляти сигнал в часовій і частотній області одночасно.

Ідея цих часово-частотних представлень полягає в тому, щоб скоротити сигнал до декількох частин, а потім проаналізувати деталі окремо. Зрозуміло, що аналіз сигналу таким чином дасть більше інформації про те, коли і де різні частотні компоненти мали місце, але це також призводить до фундаментальної проблеми: як розбити сигнал?

Припустимо, що ми хочемо точно знати всі частотні компоненти, присутні в певний момент часу. Проблема полягає в тому, що різання сигналу відповідає згортку між сигналом та різучим вікном. Оскільки згортка в часовій області ідентична множення в частотній області і Фур'є перетворення імпульсу Дірака містить всі можливі частоти, частотні компоненти сигналу будуть розмиті над всією частотною віссю. Фактично ця ситуація є протилежною стандартному перетворенню Фур'є, оскільки ми зараз маємо роздільну здатність у часі, але немає жодної частотної різниці [11].

Основний принцип явищ, про який йдеться вище, - принцип невизначеності Гейзенберга, який, стверджує, що неможливо точно визначити частоту і точний час виникнення цього сигналу певної частоти. Іншими словами, сигнал просто не може бути представлений як точка в частотно-часовому просторі.

Принцип невизначеності показує, що дуже важливо, як розбивається сигнал.

Вейвлет перетворення або вейвлет-аналіз є, мабуть, самим останнім рішенням для подолання недоліків перетворення Фур'є. У вейвлет-аналізі використання повністю масштабованого модульованого вікна вирішує проблему розбиття сигналу.

Вікно зміщується вздовж сигналу, і для кожного положення розраховується спектр. Тоді цей процес повторюється багато разів з трохи коротшим (або довшим) вікном для кожного нового циклу. В результаті буде збір частотно-часових представлень сигналу з різною роздільною здатністю. У випадку вейвлетів ми не говоримо про часово-частотні представлення, але про часово-масштабне представлення. Великий масштаб показує загальну картину, а невеликі масштаби показують деталі.

3.3 Вейвлет-перетворення

Вейвлет-аналіз - це математична процедура, яка дозволяє скоротити дані на різній частоті компонентів, а потім вивчати кожен компонент з роздільною здатністю, відповідно до її масштабу. У вейвлет-аналізі, сигнал розкладається на його "вейвлет", масштабовану та зміщену версію оригіналу (материнський) вейвлет. Взагалі, вейвлет перетворення поділяють на дискретне вейвлет перетворення (DWT), вейвлет-перетворення пакетів (WPT) і безперервне вейвлет-перетворення (CWT). Дискретне вейвлет перетворення (DWT) можна ефективно реалізувати, розбиваючи сигнал на наближуючі (низька частота) та деталізуючі (висока частота) коефіцієнти. Після першого рівня, наближення розкладається на більш високий рівень. Для вейвлет-перетворення пакетів (WPT), після першого рівня, деталі та наближення розкладаються на наступний рівень. На відміну від дискретного вейвлет перетворення, CWT може працювати у кожному масштабі. Під час обчислення, вейвлет масштабується і зміщується над повний домен аналізованого сигналу [12].

У вейвлет-аналізі існує безліч типів материнських вейвлетів, які можуть бути використані для вейвлет аналізу. Материнські вейвлети характеризуються такими властивостями, як ортогональність, компактна підтримка, симетрія і зникаючий момент. Різні материнські вейвлети, що використовуються для аналізу того ж сигналу, дадуть різні результати.

3.4 Неперервне вейвлет-перетворення

З доступних вейвлет перетворень, SWT, можливо, найпростіший для візуалізації. SWT схоже на перетворення Фур'є, де довільна функція часу може бути представлена нескінченним підсумовуванням синусоїдальних функцій та їх мультиплікативними коефіцієнтами. При вейвлет-аналізі синусоїдальні функції замінюються функціями вейвлета. Суть розрахунку полягає в тому, щоб визначити коефіцієнти, необхідні для точного зображення функції часу. Вейвлет - це математична функція, зазвичай часу. Терміни "вейвлет" та "вейвлет функція" є взаємозамінними. Вейвлет - це невелика хвиля: вона має бути коливальною і обмеженою. Використання локалізованих у часі функцій позбавляє вимог до вікон, які часто зустрічаються в методах Фур'є. Проте термін "вікна" як і раніше є загальним у вейвлет-аналізі. Ефекти витоків, від якого постраждало перетворення Фур'є, викликані через вікно, відсутні у вейвлет-перетворенні, і тому інформація щодо сигналу зберігається. Вейвлет повинен відповідати ряду математичних умов, які обмежують типи функцій, які можуть бути використані при аналізі. Три приклади вейвлет-функцій показані на рисунку 3.1.

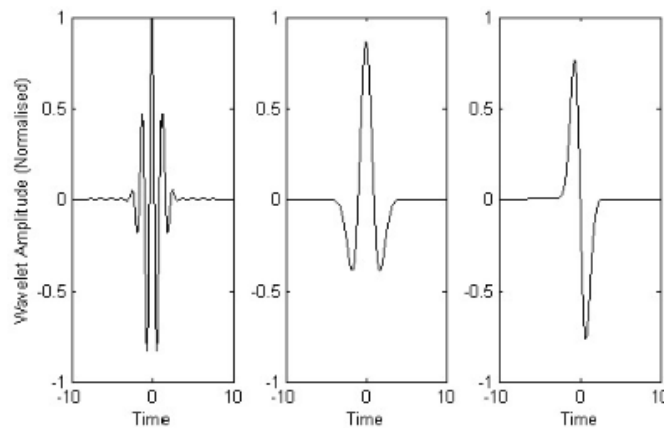


Рисунок 3.1 – Приклади вейвлетів: мортлет, мексиканський капелюх, гаусіан

Визначення вейвлету вимагає розуміння процесу масштабування та переміщення. Щоб масштабувати вейвлет, це означає розтягувати або розширити: це гарантує, що енергія, що міститься в масштабованих вейвлетах, така ж, як у вихідного материнського вейвлету. Коли вейвлет розтягнутий у горизонтальному напрямку в

осі, він вирівнюється у вертикальному напрямку осі. Переміщення вейвлету, як правило, виконується у позитивному напрямку вздовж осі x . У більшості випадків вісь x буде представляти час, а ось y - амплітуда. На практиці горизонтальна вісь не завжди "час", але завжди буде містити різноманітну інформацію часу. Процес перетворення досягається завдяки постійному масштабування та переміщенню материнського вейвлету по довжині сигналу. Перед здійсненням перетворення необхідно вибрати функції вейвлету. Термін "масштаб" часто використовується щодо частоти функції вейвлету. Масштаб обернено пропорційний частоті, тому низька шкала відноситься до самих щільно упакованих (високочастотних) вейвлетів. Наслідком цього є те, що типовими вейвлет-розкладаннями є функція часу і масштабу, а не час і частота. Вейвлет перетворення зазвичай починають використовувати низькочастотні (високочастотні) вейвлет-функції, що проходять до високих масштабів (низька частота), де вейвлет є найбільш розширеним. Для початку аналізу, вейвлет встановлюється на початку сигналу $t=0$. Результат вейвлету та сигналу потім інтегрується протягом усього часу. Результати потім нормалізуються, даючи вейвлет-коефіцієнт для цього масштабу та переміщення. Вейвлет потім рухається з деяким невеликим приростом в позитивному напрямку часу і розраховується новий коефіцієнт, цей процес продовжується до досягнення кінця сигналу. У цьому пункті шкала збільшується, вейвлет розтягується і розширюється, а сигнал аналізу повертається до відправної точки, тобто $t = 0$, що дозволяє обчислити коефіцієнти вейвлетів у наступному масштабі. Трансформація завершується, коли сигнал аналізується для всіх масштабів [13].

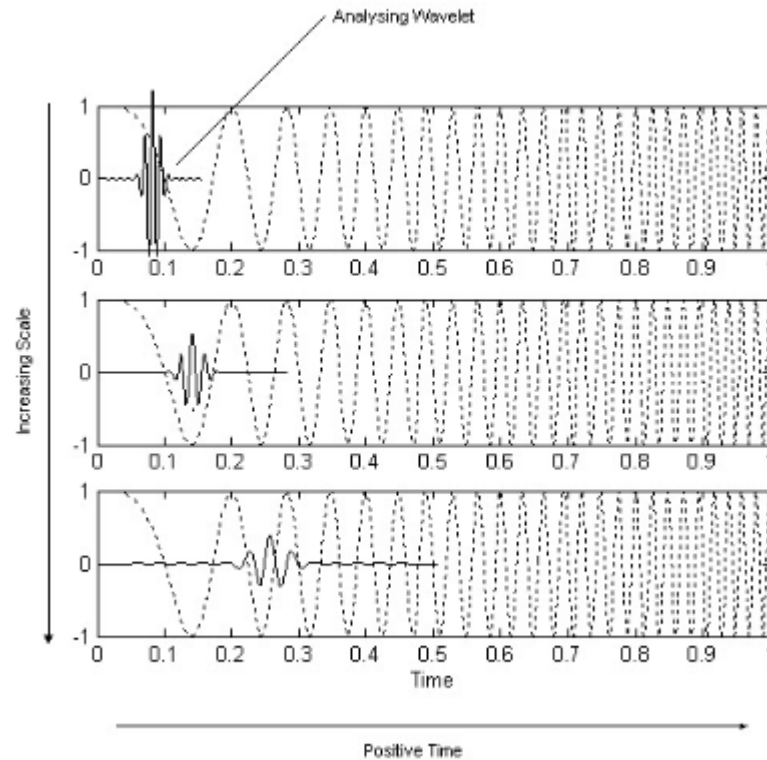


Рисунок 3.2 – Неперервне вейвлет-перетворення

Вейвлет-перетворення виконує порівняння вейвлет-сигналу. Існує висока ступінь подібності між двома функціями, коли коефіцієнти при перекладі та масштабі великі. Якщо два невідповідні, коефіцієнти невеликі. Таким чином, процес перетворення дає вказівки частотного змісту сигналу. Оскільки приблизний зв'язок існує між масштабом та частотою, великий коефіцієнт у певній шкалі передбачає наявність певної частоти. Здатність перетворення вейвлетів надати точну інформацію про час і частоту обмежується принципом невизначеності так само, як перетворення Фур'є. Просто це означає, що одночасно неможливо досягти як гарного часу, так і доброго частотного вирішення. Це чітко продемонстровано за допомогою перетворення Фур'є, коли взагалі немає інформації про час через нескінченне вікно, що передбачено в техніці. STFT забезпечує роздільну здатність у часі шляхом зменшення цього вікна: у міру зменшення вікон поліпшується часовий розряд за рахунок роздільної здатності частоти.

Аналіз різних масштабів у вейвлет-перетворенні забезпечує гнучке розбиття на вікна. Отримане перетворення матиме різну роздільну здатність часу/частоти. Малі

масштаби(більш високі частоти) займають невеликі вікна, де вейвлет дуже компактний. Оскільки розмір вікна зменшується, часова роздільна здатність поліпшується за рахунок роздільної здатності частоти (як у STFT). Більші масштаби(нижча частота) розтягуються і займають більші вікна, таким чином, вони мають більш низьку роздільну здатність часу і кращу роздільну здатність частот (знову ж таки, це схоже на STFT з великими вікнами). Гнучка схема, прийнята вейвлет-перетвореннями, забезпечує компроміс між частотною та часовою роздільною здатністю. Це зазвичай виявляється вигідним при аналізі більшості сигналів. Реальні сигнали, як правило, демонструють високочастотні компоненти для коротких термінів і низькочастотних компонентів протягом тривалого часу.

3.5 Дискретне вейвлет-перетворення

Попередній підрозділ розглядав безперервне вейвлет-перетворення. Це забезпечило демонстрацію деяких ключових понять, що беруть участь у вейвлет-аналізі. CWT - алгоритм, що вимагає масштабних обчислень, так що, як і з використанням Фур'є, розроблено дискретне вейвлет-перетворення (DWT). DWT відрізняється від CWT, оскільки метод обчислення використовує кодування піддіапазону. Оскільки в даних, що містяться в CWT, є велика кількість непотрібних даних, DWT використовує вибірку як даних масштабу, так і часу, завдяки чому виробляється значно швидший алгоритм [14]. Масштаб та час відбираються у степенях двійки. Дані DWT, як правило, відображаються як серія ділянок, де кожен ділянка представляє певний масштаб. На рисунку 3.3 показано, як діаграми DWT відносяться до діаграми CWT. Материнським, у даному випадку, було обрано вейвлет Добеши 7, відомий як 'db7' (рисунк 3.4). Аналізуючи дві діаграми одночасно, можна побачити, що частина високих амплітуд сигналів відноситься до світлих ділянок в CWT. Необхідна інформація, представлена в CWT, може бути повністю представлена на шести ділянках DWT.

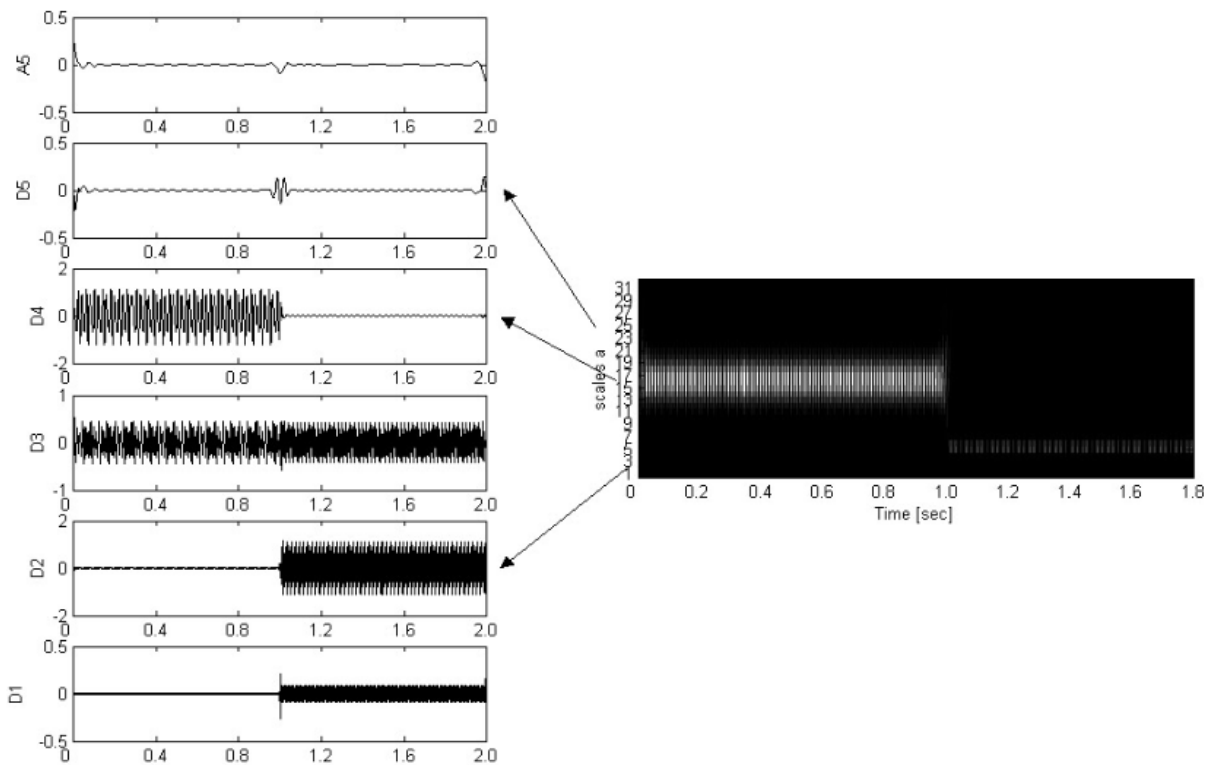


Рисунок 3.3 – DWT та CWT використовуючи вейвлет Добеши 7

Значення осі Y - "A" для наближення та "D" для деталі; D1 - низький масштаб з збільшенням чисел у вищих масштабах. Апроксимація зазвичай показує основну тенденцію даних: компоненти низької частоти. У цьому прикладі це більш-менш пряма лінія. Решта рівнів деталізації показують, як сигнал змінюється між зразками часу. DWT містить ту саму важливу інформацію, що й CWT. Рівень D3, наведений на рисунку 3.3, відсутній у CWT за рахунок прийняття іншої функції вейвлету.

CWT на рисунку 3.2 аналізується більш ніж на 32 масштабах, тому DWT потрібно аналізувати на 5 рівнях : $2^1, 2^2, \dots, 2^5$ дає зразки масштабів 2, 4, ..., 32. Ці п'ять рівнів являють собою деталізовані схеми, показані на малюнку 3.3, де рівень D1 відповідає масштабу 2 (2^1), D2 до масштабу 4 (2^2), D3 до масштабу 8 (2^3) і т. д. DWT відрізняється від CWT в тому, що він робить використання функції масштабування на додачу до функції вейвлету, яку раніше обговорювали; це є причиною додаткової діаграми.

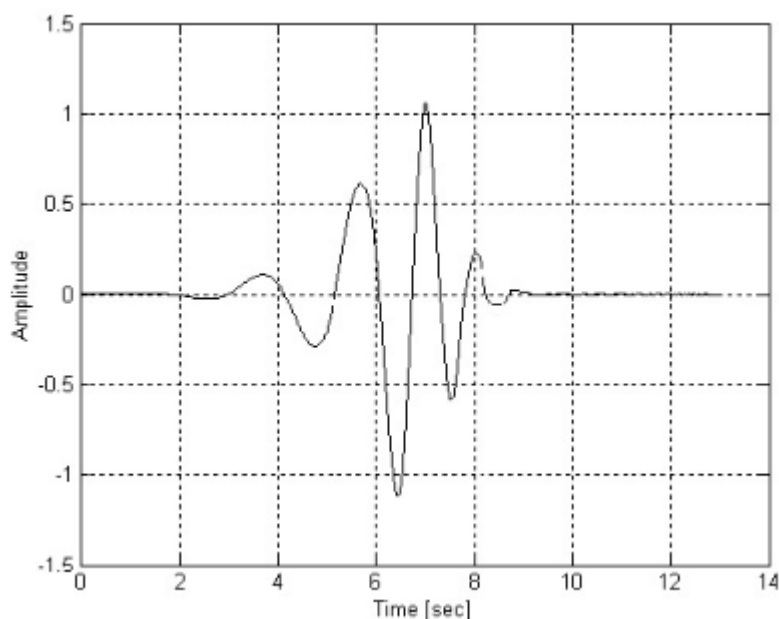


Рисунок 3.4 – Вейвлет Добеши 7

У DWT вейвлет-функція використовується для визначення коефіцієнтів деталізації, а функція масштабування визначає коефіцієнти наближення. Кожен рівень розкладання видаляє більше високочастотних даних з коефіцієнтів наближення [15]. У розглянутому випадку, лише невелика кількість низькочастотних даних була збережена, залишаючи загальну тенденцію сигналу. Щоб охопити весь спектр частот від нуля до верхньої межі, буде потрібно необмежене число масштабованих вейвлет-функцій. Рішенням є створення функції з низькочастотними властивостями з використанням функції масштабування. Таким чином, DWT розраховується шляхом пропускання сигналу через фільтр-банк, що складається з фільтра високих і низьких частот. На кожному проході через банк фільтрів функція масштабування фіксує дані низької частоти з попереднього наближення. Ефект подвійного семплінгу від масштабованої функції полягає в тому, щоб поступово зменшувати компоненти низької частоти, що містяться в кожному наступному наближенні [16].

За допомогою DWT, сигнал $x(t)$ можна представити формулою:

$$x(t) = \sum_n c_{m-k,n} \Phi(2^{m-k} t - n) + \sum_{i=0}^k \sum_n d_{m-i,n} \Psi(2^{m-i} t - n), \quad (1)$$

де $c_{m-k,n}$ - апроксимуючі коефіцієнти, $d_{m-i,n}$ - деталізуючі коефіцієнти, k – рівень декомпозиції, n – номер семплу, Ψ - вейвлет, Φ - масштабуюча функція.

3.6 Зворотне дискретне вейвлет-перетворення

Зворотне дискретне вейвлет-перетворення (IDWT) дозволяє передавати сигнали від системи вимірювання співвідношення часових масштабів до системи часових значень. Вейвлет-перетворення є перетворенням без втрат - вся енергія зберігається під час переходу сигналу з часового домену в вейвлет-коефіцієнти і навпаки. Рівність, що визначає зворотне дискретне перетворення [17]:

$$f(t) = \sum_{m,n} (f, \psi_{mn}) \psi_{mn} = \sum_m \sum_n d_m[n] \psi_{mn}, \quad (2)$$

У вищезгаданій формулі вейвлет-коефіцієнти $d_m[n] = (f, \psi_{mn})$ визначають загальні риси сигналу f та вейвлет ψ_{mn} . Чим більше значення вейвлетного коефіцієнта, тим більше кореляція (ступінь відповідності) між сигналом вейвлету сигналу та вейвлетом, з довжиною визначеною m та позицією визначеною n і t .

3.7 Фільтрація сигналу та обчислення порогового значення

Фільтрація сигналу відбувається у деталях розбитого сигналу. Це передбачає очищення всіх елементів сигналу, які не відповідають заданому стану [18]. У цьому випадку умова визначається функцією з вибраного порогу. Поріг може бути розрахований на основі сигналу або адаптований індивідуально. Як правило, у порозі ліквідації використовується жорсткий або м'який поріг. Вищезгадані методи часто використовуються через простоту розрахунків. На жаль, для певних цілей ці пороги можуть бути недостатніми через їх несприятливі властивості [19]. Виключення жорсткого порогу визначається виразом:

$$\delta_\lambda(d) = \begin{cases} d, & |d| > \lambda \\ 0, & |d| \leq \lambda \end{cases} \quad (3)$$

М'який поріг усунення, що вирішує проблему з розривом функцій, визначається формулою:

$$\delta_{\lambda}(d) = \begin{cases} 0, & |d| \leq \lambda \\ d - \lambda, & d > \lambda \\ d + \lambda, & d < -\lambda \end{cases} \quad (4)$$

Приклади м'якого та жорсткого порогового значення продемонстровані на рисунках 3.5 та 3.6.

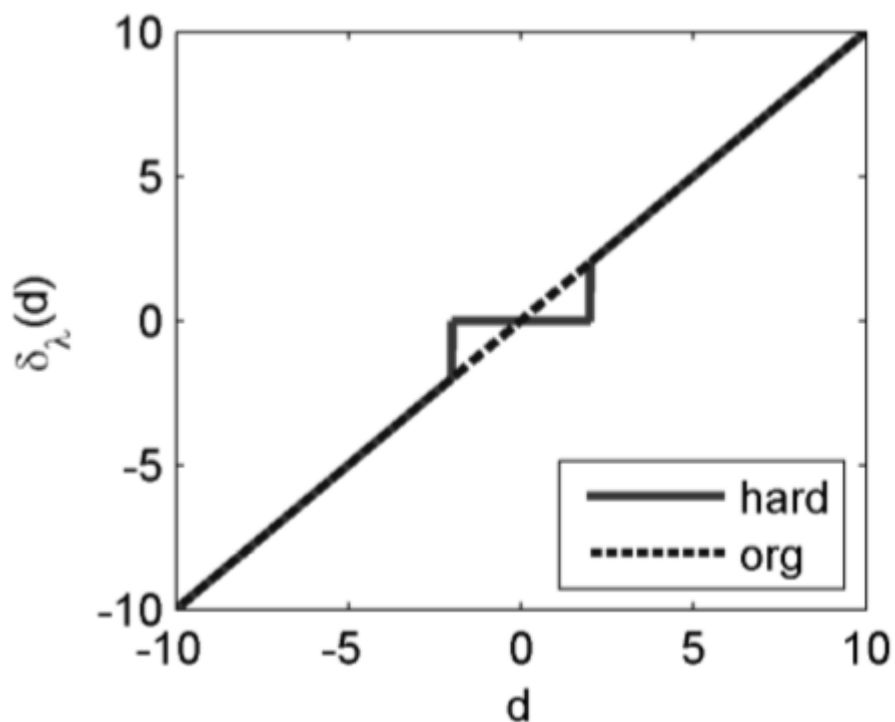


Рисунок 3.5 – Приклад жорсткого порогового значення

Є три типи порогу: глобальний поріг, місцевий поріг або місцевий пороговий розподіл [20]. Глобальний поріг приймає значення λ на кожному рівні наближення, як тільки будуть відомі параметри при вхідному сигналі. Використання такого порогу досить зручне на практиці, але менш ефективне. Місцевий поріг передбачає інше значення для кожного рівня розкладання. Розмір порогу розраховується на основі коефіцієнтів деталей. Локальний пороговий розподіл передбачає різні значення для

фіксованих коефіцієнтів деталізації. Іншими словами, для деталей в діапазоні $< d_1; d_2 >$ використовується інший поріг, ніж для діапазону $< d_3; d_4 >$.

Рисунок 3.6 – Приклад м'якого порогового значення

Одним із найпоширеніших способів визначення порогу є використання універсального порогу, де порогове значення суворо залежить від нормалізованого стандартного відхилення деталей на кожному рівні розкладання [21].

$$\lambda = \sqrt{2 \log n} \frac{\sigma}{\sqrt{n}}, \quad (5)$$

$$\sigma^2 = E\{[d - E(d)]^2\} = \sum_{i=1}^n (d_i - \bar{d})^2, \quad (6)$$

де n – номер семплу, d – деталізуюче значення, σ - стандартне відхилення деталі.

Для проведення фільтрації гідроакустичного сигналу потрібно виконати наступні кроки:

- Розкласти сигнал за допомогою DWT.
- Відфільтрувати сигнал у вейвлет-просторі, використовуючи порогове значення.

- Інвертувати відфільтрований сигнал для реконструкції оригінального, тепер відфільтрованого сигналу, використовуючи зворотний DWT.

Таким чином, схема проведення фільтрації зображена на рисунку 3.7.

Рисунок 3.7 – Схема фільтрації сигналу.

4 ТЕХНОЛОГІЧНА БАЗА РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАСОБУ

Програмний засіб розроблений з використанням мови програмування С# на платформі .Net Framework 4.6.2. Для розробки графічного інтерфейсу було використано технологію WPF. Для представлення результатів у вигляді графіків було використано бібліотеку Live Charts [22]. Для виконання математичних розрахунків до проекту було додано бібліотеку NMath [23].

4.1 Основні відомості про мову програмування С#

На сьогоднішній момент мова програмування С# одна із найпотужніших та найбільш затребуваних мов програмування у сфері інформаційних технологій. Дана мова дозволяє створювати найрізноманітніші проекти: від простих мобільних додатків, до високонавантажених веб-порталів.

С# відноситься до сімейства мов із С-подібним синтаксисом, тому синтаксично схожий на такі відомі мови, як Java та С++. Це об'єктно-орієнтовна мова програмування з відносно строгою типізацією, оскільки в останніх версіях присутні деякі можливості динамічної типізації. Підтримує перевантаження операторів, делегати, події, анонімні функції, замикання, атрибути, узагальнені типи та методи, ітератори. [24]

Переїнявши багато у своїх попередників – мов С, С++, Smalltalk, Java, опираючись на різні методики та підходи до розробки, ця мова переїняла краще та виключила деякі моделі, які зарекомендували себе як проблематичні, наприклад, в С#, на відміну від С++, не підтримується багаторазове наслідування класів, проте присутнє багаторазове наслідування інтерфейсів.

С# продовжує активно розвиватись. З кожною новою версією з'являється все більше цікавих можливостей, як, наприклад, асинхронні методи, лямбда вирази, динамічне зв'язування тощо.

4.2 Концепція .NET

Вважається, що платформа .NET Framework стала відповіддю компанії Microsoft на велику популярність, що набрала на той час, платформу Java. .NET є схожим набором стандартів, інструментів і бібліотек, але розробка застосувань у рамках .NET можлива з використанням різних мов програмування. Деякою відмінністю від Java також є те, що код (на проміжній мові) в .NET не інтерпретується, а завжди виконується в режимі динамічної компіляції (JIT). Розробка платформи почалася в другій половині 1990-х. Офіційно про розробку нової технології було оголошено 13 січня 2000 року, в день коли Біл Гейтс офіційно оголосив про передачу поста глави Microsoft Стіву Балмеру. Цього дня керівництвом корпорації була озвучена нова стратегія компанії, що дістала назву Next Generation Windows Services (скор. NGWS, Нове покоління служб Windows). Нова стратегія повинна була об'єднати в єдиний набір існуючі і майбутні розробки Microsoft для надання можливості користувачам працювати зі Всесвітньою павутиною з безпроводних пристроїв, що мають доступ в Інтернет, як із стаціонарних комп'ютерів. .NET Framework не містить особливих вказівок стосовно типів застосувань, які можуть створюватися за допомогою цієї платформи. Пояснюється це тим, що ніяких обмежень подібного роду просто не існує: .NET Framework дозволяє створювати застосування для Windows, Web-застосування, застосування типу Web-служб і практично усі інші типи застосувань, які тільки можна собі уявити. Окрім цього ще існує і версія Microsoft .NET Compact Framework, яка, по суті, є усіченим варіантом повної версії .NET Framework і може застосовуватися на пристроях типу персональних цифрових помічників (personal digital assistant - PDA) і навіть в деяких смартфонах. Доступна також .NET Micro Framework – це реалізація платформи Microsoft .NET для вбудовуваного застосування в 32- і 64-розрядних мікроконтролерах. Нині реалізована на мікроконтролерах з архітектурою ARM7, ARM9 і Blackfin. Не вимагає наявності ОС. Одним з головних стимулів до застосування платформи .NET Framework являється можливість використати її в якості засобу інтеграції різних операційних систем. Компанія Microsoft ініціювала прийняття стандартів, що описують її окремі елементи (на жаль, поки не всі), і вона

ж є основним постачальником реалізацій цієї платформи і інструментів розробки. Завдяки наявності стандартів можлива незалежна реалізація .NET (наприклад, така реалізація розроблена у рамках проекту Mono), але, в силу молодості платформи (спочатку) і побоювань з приводу монопольного впливу Microsoft на її подальший розвиток, реалізації .NET не від Microsoft використовуються досить рідко.

4.3 Архітектура .NET Framework

Основою .NET являються віртуальна машина для проміжної мови (Intermediate Language — IL, іноді зустрічається скорочення Microsoft IL — MSIL, пізніше прийняли назву Common Intermediate Language — CIL), в яку транслюються усі .NET-програми, що також називається загальним середовищем виконання (Common Language Runtime — CLR), і загальна бібліотека класів (.NET Framework class library, FCL), доступна з усіх .NET-застосувань. Проміжна мова є повноцінною мовою програмування, але вона не призначена для використання людьми. Розробка у рамках .NET ведеться на одній з мов, для яких є транслятор, в проміжну мову — Visual Basic.NET, C++, C#, Java (транслятор Java в .NET називається J#, і він не забезпечує однакової роботи програм на Java, що відтрансльовані в .NET і виконуваних на JVM) і ін. Проте різні мови досить сильно відрізняються одна від одної, і щоб гарантувати можливість з однієї мови працювати з компонентами, написаними на іншій мові, необхідно при розробці цих компонентів дотримуватися загальних правил (Common Language Specifications — CLS), які визначають, якими конструкціями можна користуватися в усіх .NET-мовах без втрати можливості взаємодії між результатами. Найбільш близький до проміжної мови C# — ця мова була спеціально розроблена разом з платформою .NET.

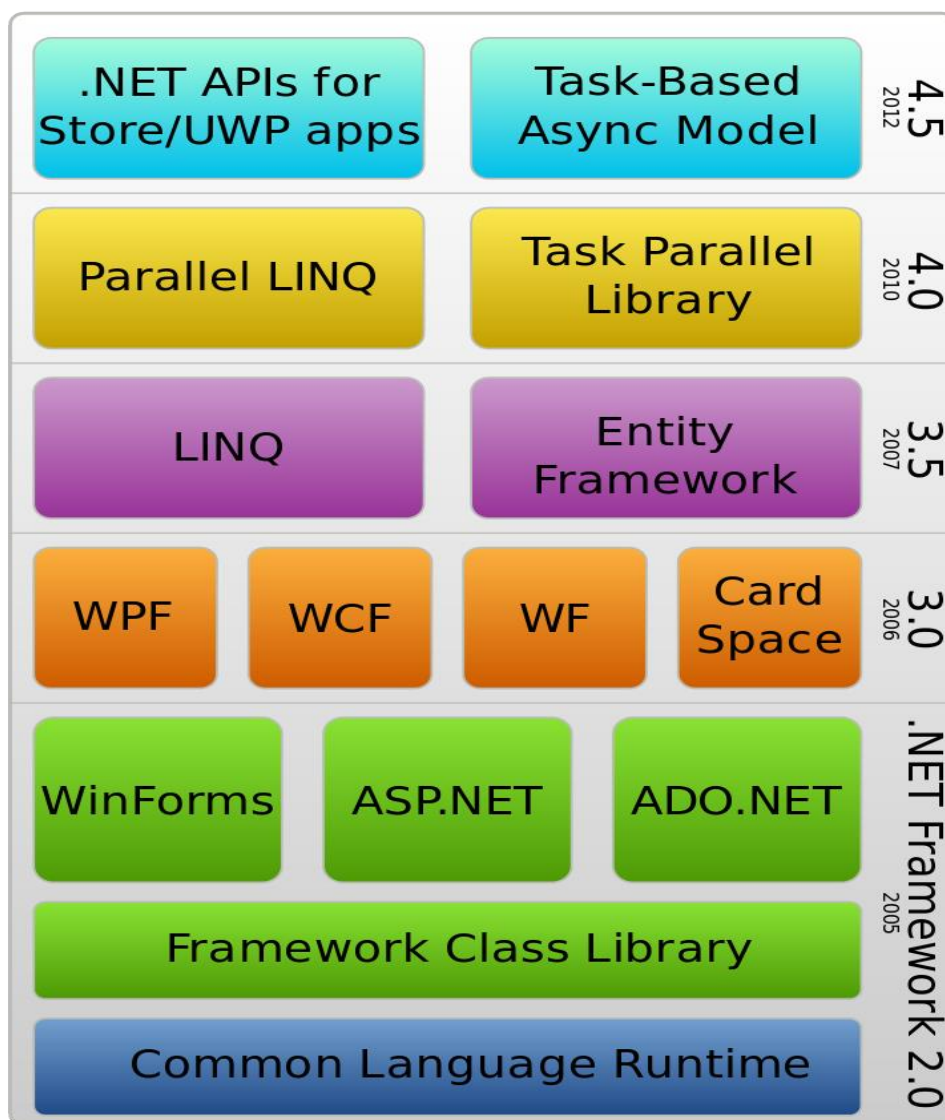


Рисунок 4.1 – Архітектура .Net Framework

При компіляції коду, в якому використовується бібліотека .NET, він не перетворюється відразу ж в рідний код конкретної операційної системи. Замість цього він спочатку перетвориться в код MSIL/CIL. Цей код не є специфічним ні для якої-небудь операційної системи, ні для мови C#. Інші мови, наприклад, Visual Basic .NET, на першому етапі теж компілюються в код на цій мові. У разі розробки застосувань на C# такий процес компіляції виконується VS. Очевидно, що далі для запуску застосування вимагається виконати ще деяку роботу. За це відповідає так званий JIT-компілятор (Just-in-Time compiler – оперативний компілятор), який компілює MSIL/CIL в рідний код, що відповідає вимогам конкретної операційної системи і архітектури цільового комп'ютера. Тільки після цього етапу у операційної системи

з'являється можливість запустити застосування. Аббревіатура JIT в назві компілятора вказує на те, що він виконує компіляцію CIL-коду тільки при виникненні відповідної необхідності.

4.4 Windows Presentation Foundation

В основі WPF лежить незалежний від дозволу векторний модуль візуалізації, що використовує можливості сучасного графічного обладнання. Можливості цього модуля розширюються за допомогою комплексного набору функцій розробки додатків, які включають в себе мову XAML, елементи управління, прив'язку до даних, макет, двовимірну і тривимірну графіку, анімацію, стилі, шаблони, документи, мультимедіа, текст і типографічні функції. WPF входить до складу .NET Framework, тому ви можете створювати додатки, що включають інші елементи бібліотеки класів .NET Framework [25].

Технологія WPF (Windows Presentation Foundation) є частина екосистеми платформи .NET і являє собою підсистему для побудови графічних інтерфейсів.

Якщо при створенні традиційних додатків на основі WinForms за відрисовку елементів управління і графіки відповідали такі частини ОС Windows, як User32 і GDI+, то додатки WPF засновані на DirectX. У цьому полягає ключова особливість рендеринга графіки в WPF: використовуючи WPF, значна частина роботи по відображенні графіки, як найпростіших кнопочок, так і складних 3D-моделей, лягати на графічний процесор на відеокарті, що також дозволяє скористатися апаратним прискоренням графіки.

Однією з важливих особливостей є використання мови декларативною розмітки інтерфейсу XAML, заснованого на XML: ви можете створювати насичений графічний інтерфейс, використовуючи або декларативне оголошення інтерфейсу, або код на керованих мовах C# і VB.NET, або поєднувати і те, і інше.

Особливості WPF:

- Використання традиційних мов .NET-платформи - C# і VB.NET для створення логіки додатка.

- Можливість декларативного визначення графічного інтерфейсу за допомогою спеціальної мови розмітки XAML, заснованому на xml і представляє альтернативу програмному створенню графіки та елементів управління, а також можливість комбінувати XAML і C # / VB.NET.

- Незалежність від дозволу екрану: оскільки в WPF всі елементи вимірюються в незалежних від пристрою одиницях, додатки на WPF легко масштабуються під різні екрани з різним дозволом.

- Нові можливості, яких складно було досягти в WinForms, наприклад, створення тривимірних моделей, прив'язка даних, використання таких елементів, як стилі, шаблони, теми і ін.

- Хорошу взаємодію з WinForms, завдяки чому, наприклад, в додатках WPF можна використовувати традиційні елементи управління з WinForms.

- Багаті можливості по створенню різних додатків: це і мультимедіа, і двовірна і тривимірна графіка, і багатий набір вбудованих елементів управління, а також можливість самим створювати нові елементи, створення анімацій, прив'язка даних, стилі, шаблони, теми і багато іншого.

- Апаратне прискорення графіки - незалежно від того, чи працюєте ви з 2D або 3D, графікою або текстом, всі компоненти програми транслюються в об'єкти, зрозумілі Direct3D, і потім візуалізуються за допомогою процесора на відеокарті, що підвищує продуктивність, робить графіком більш плавною.

- Створення додатків під безліч ОС сімейства Windows - від Windows XP до Windows 10.

У той же час WPF має певні обмеження. Незважаючи на підтримку тривимірної візуалізації, для створення додатків з великою кількістю тривимірних зображень, перш за все ігор, краще використовувати інші засоби - DirectX або спеціальні фреймворки, такі як Monogame або Unity.

Також варто враховувати, що в порівнянні з додатками на Windows Forms обсяг програм на WPF і споживання ними пам'яті в процесі роботи в середньому трохи вище. Але це з лишком компенсується більш широкими графічними можливостями і підвищеною продуктивністю при відображенні графіки.

4.5 Model-View-ViewModel

У розробці програмного забезпечення існує безліч моделей для того, як структурувати ваш код. MVVM або Model-View-ViewModel - це шаблон розробки, який ґрунтується навколо розділення обов'язків у додатках WPF (і Silverlight) [26]. Для цього додатки поділяються на три основні компоненти: View, ViewModel та Model.

Розглянемо компонент View. Він несе відповідальність за те, що буде відображатися користувачеві. У цьому світлі View можна розглядати як шаблон, який відображатиме певну інформацію для користувача та оброблятиме взаємодію користувача в програмі. Це може бути форма подання, пошукова система аукціону або будь-яка інша кількість можливих сценаріїв для користувача. У багатьох випадках цей View буде оброблений дизайнером, який спеціалізується на UX (досвід користувача), але мало знає про домен або бізнес-логіку, що йде в додаток. З огляду на це, ви не хочете, щоб ви безпосередньо писали складний бізнес-код у View. Замість цього ви хочете, щоб його абстрагували, щоб думка та бізнес-логіка могли працювати самостійно та об'єднуватися через багату інфраструктуру в WPF та Silverlight.

Це приводить нас до ViewModel, справжню робочу частину програми, яка містить бізнес-логіку, тобто основну частину програми. Через зв'язування даних, загальнодоступних властивостей та колекцій, ViewModel може бути виставлено для перегляду, і навіть може бути змінений в залежності від типу зв'язування. Це дозволяє чітко розділити представлення та логіку програми.

У багатьох сценаріях Model буде витягуватися з системи ORM, як-от Telerik Data Access або Entity Framework. Незалежно від того, звідки надходять дані, поки ViewModel викликає правильні методи у вас є повний та цілісний додаток. Коли ці три елементи об'єднати, то ви отримаєте повністю функціональне додаток. View обробляє взаємодію з користувачем та показує інформацію користувачеві, Model відповідає за дані програми та різні операції CRUD та операції запиту, які

виконуються, і ViewModel існує як клей, який поєднує ці два з допомогою методу набору даних Silverlight та WPF двигун

У світі, де ми маємо можливість перетягувати комплексний елемент управління з нашої панелі інструментів, провести декілька дій та отримати швидку функціональність, а потім перейти до наступного робочого пункту, ми іноді забуваємо, що просте розробка програм не обов'язково прирівнюється до програмного забезпечення, яке добре структуроване, легко розширюване, а його роботу можна перевірити. MVVM допомагає цим всім.

Розвиваючи за допомогою моделі MVVM, існує чітке розділення між тим, де повинен існувати певний код. Оскільки View несе повну відповідальність за презентацію, він не містить коду (крім випадків, коли код дуже специфічний для View і відповідає за відображення даних та збір відгуків. ViewModel містить логіку, доступну через зв'язування, але ніколи не буде безпосередньо комунікувати із View. Насправді, ViewModel абсолютно не знає, який вигляд буде підключено до нього. Тому, якщо функціональність не працює, як очікувалося, ви можете швидко перевірити представлення, щоб забезпечити відповідність тверджень; інакше ваш ViewModel буде сегментовано та розділено, і його можна протестувати окремо від інтерфейсу користувача.

Такий підхід дозволяє нам тестувати одиницю. У цих випадках автоматичне тестування вступає в гру, і його можна використовувати для забезпечення того, щоб ваш інтерфейс залишався функціональним, коли ви працювали над своєю програмою. Але перш ніж щось натискається на користувацькому інтерфейсі, ви хочете отримати достатню впевненість у тому, що ваша програма буде працювати, як очікувалося, і можна провести тестування коду. Щоб протестувати кожен з трьох компонентів окремо, вам доведеться виділити їх залежності до шару нижче. Наприклад, якщо ви хочете протестувати ViewModel, вам доведеться якось заповнити модель, щоб перевірити поведінку ViewModel. У таких випадках використовують Mock фреймворки, які допомагають полегшити цей процес.

Іншою причиною для розробки в більш модульному, сегментованому підході, як MVVM, є поєднання розширюваності та ремонтпридатності. За допомогою

стандартних методів ви часто закінчуєтеся захаращеним безладом коду, який важко читати, і ще важче його налагодити. За допомогою моделі MVVM у вас є чіткий розподіл, що полегшить даний процес.

4.6 XAML

XAML - це розширювана мова розмітки додатків. Це проста мова на основі XML для створення та ініціалізації об'єктів .NET з ієрархічними відносинами [27]. Хоча він був спочатку винайдений для WPF, він може, використовуючи для створення будь-яких дерев об'єктів.

Сьогодні XAML використовується для створення користувацьких інтерфейсів у WPF, Silverlight, оголошення робочих процесів у WF.

Всі класи у WPF мають конструктори без параметрів та використовують властивості. Це зроблено, щоб зробити його ідеально підходящим для мов XML, таких як XAML.

Все, що ви можете зробити в XAML, також можна зробити за допомогою коду. XAML - це ще один спосіб створити та ініціалізувати об'єкти.

Ви можете використовувати WPF без використання XAML. Це залежить від вас, чи ви хочете оголосити його в XAML або написати його в коді. Використання XAML має деякі переваги:

- Код XAML короткий і простий для читання.
- Відокремлення дизайнерського коду та логіки.
- Інструменти для графічного проектування, такі як Expression Blend, вимагають використання XAML як джерела.
- Відокремлення логіки XAML та інтерфейсу користувача дозволяє чітко відокремити ролі дизайнера та розробника.

Також для спрощення роботи з XAML було створено розширення розмітки - це динамічні заповнювачі для значень атрибутів. Вони вирішують значення властивості під час виконання. Розширення розмітки надруковані фігурними дужками (наприклад: `Background="{StaticResource NormalBackgroundBrush}"`). WPF має деякі

вбудовані розширення розмітки, але ви можете писати свої власні, виводячи з MarkupExtension. Вбудовані розширення розмітки:

- Binding.
- StaticResource.
- DynamicResource.
- TemplateBinding.
- x:Static.
- x:Null.

Першим ідентифікатором в парі фігурних дужок є назва розширення. Всі перехресні ідентифікатори називаються параметрами у формі Property = Value.

4.7 Dependency injection

Dependency injection або впровадження залежностей – механізм, що дозволяє забезпечити слабку зв'язність компонентів системи [28]. Такі компоненти зв'язані між собою через абстракції, наприклад через інтерфейси. При цьому відповідальність за створення відповідних залежностей покладається на зовнішній, спеціально призначений для цього загальний механізм.

Часто, в ролі таких механізмів виступають ІоС-контейнери. Такі контейнери служать своєрідними фабриками, які встановлюють залежність між абстракціями та конкретними реалізаціями і, як правило, керують створенням цих об'єктів.

Такий дозволяє збільшити гнучкість системи, полегшити підтримку, розширення та заміну компонентів системи.

4.8 Середовище розробки Visual Studio

Microsoft Visual Studio – серія програмних продуктів компанії Microsoft для розробки програмного забезпечення. Включає в себе інтегроване середовище розробки та перелік інших засобів для розробки та впровадження програмного забезпечення [29].

Visual Studio включає в себе редактор коду, що підтримує технологію IntelliSense – компонент завершення коду. Інші вбудовані інструменти включають в себе: конструктор форм для побудови інтерфейсів додатків, веб-дизайнер, функціональні блоки для роботи з базою даних тощо.

Підключення додаткової функціональності за допомогою плагінів дозволяє включити підтримку систем контролю версій, додаткових редакторів, наборів інструментів для інших аспектів життєвого циклу розробки програмного забезпечення.

Присутня вбудована підтримка різних мов програмування, таких як C, C++, Visual Basic, F#, C#. Підтримку інших мов, таких як Python, Ruby, М можна забезпечити за допомогою підключення відповідних мовних служб, встановлених окремо.

Visual Studio за замовчування включає наступні компоненти:

- Visual Basic .NET, а до його появи — Visual Basic
- Visual C++
- Visual C#
- Visual J#
- Visual F# (входить до складу Visual Studio 2010)
- Visual Studio Debugger.

5 Реалізація системи

5.1 Структура проекту

Відповідно до стеку обраних технологій, систему реалізовано на базі технології WPF.

Розроблений продукт має типову структуру для проектів даного типу (рисунок 5.1). Логіка програми розбита на відповідні модулі та реалізовані за допомогою інтерфейсів та технології впровадження залежностей, що сприяє легкому розширені функціоналу.

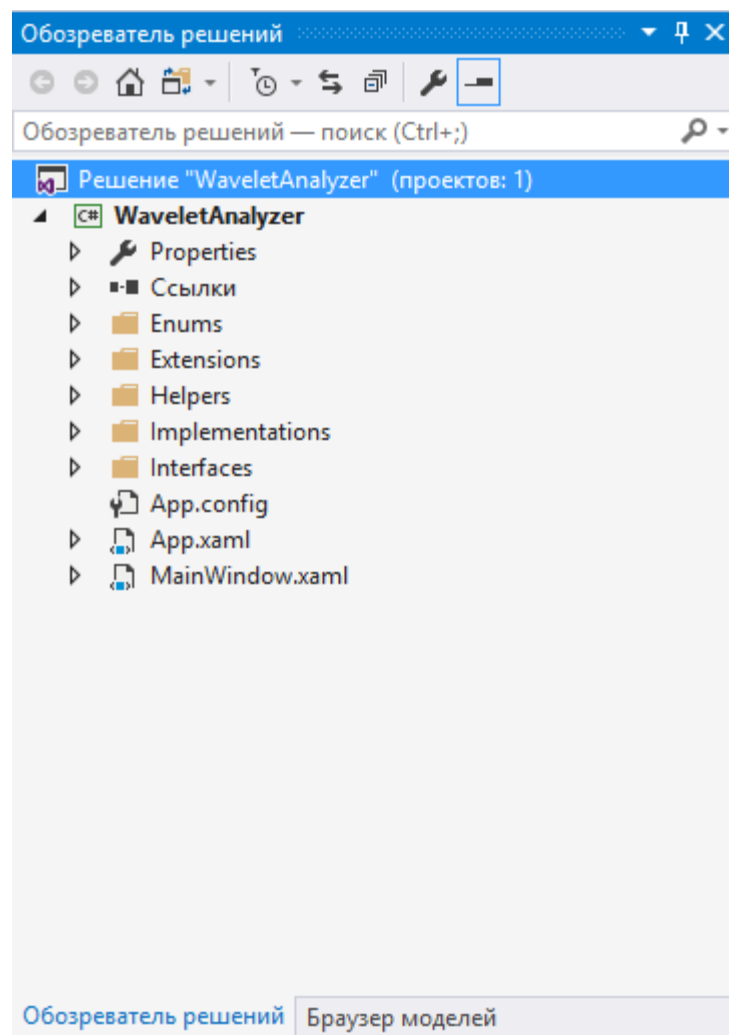


Рисунок 5.1 – Структура проекту

Структура проекту включає наступні компоненти:

- Enums – папка, що містить усі перерахування, які використовуються у розробленому програмному забезпеченні (наприклад, пороговий метод, порогова політика та інші).
- Extensions – папка, що містить методи розширення для існуючих класів (наприклад, метод розширення для List<byte>, який перетворює байтове представлення даних в числове, яке придатне для обробки).
- Helpers – папка, що містить допоміжні класи, в які винесена частина логіки, яка повторно використовуються у системі (наприклад, статичний клас WaveletHelper, який містить статичні функції для розрахунку значення вейвлета при певному наборі параметрів).
- Interfaces – папка, що містить інтерфейси, що визначають поведінку основних класів, що реалізують логіку програмного забезпечення.
- Implementations – папка, що містить класи з логікою, що реалізують вище описані інтерфейси.
- Файли конфігурації програмного засобу.
- MainWindow – клас, що описує інтерфейс та всі взаємодії з ним.

5.2 Обробка вхідних даних

Для реалізації введення та обробки вхідних даних, під час розробки програмного забезпечення, був створений відповідний інтерфейс, його базова реалізація та клас з методом розширення для List<byte>, які зображені на рисунку 5.2.

Розглянемо функціонал даного модуля:

- GetContent – відповідає за отримання даних файлу, що лежить за переданим шляхом, враховуючи багатоканальність та дискретизацію даних у файлу з гідроакустичним сигналом.
- ExtractChannels – відповідає за розбиття вхідного потоку даних на канали. Реалізація враховує той факт, що кожна точка сигналу описується двома байтами, а також те, що дані з кожного каналу чередуються в отриманому потоці.

- Метод розширення для `List<byte>` `ToDoubleList` відповідає за перетворення даних у вигляді байтів у амплітудний вигляд.

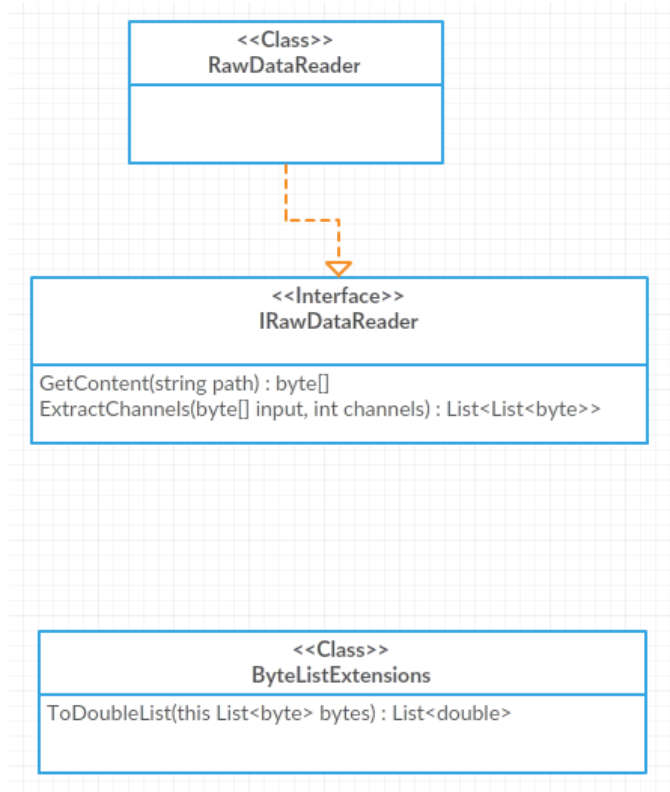


Рисунок 5.2 – Реалізація модулю обробки вхідних даних

Даний модуль повністю охоплює процес зчитування даних та їх приведення до вигляду, який можна обробити. В подальшому, за необхідності (наприклад, завантажуваний файл буде містити заголовки), функціонал можна розширити шляхом наслідування та реалізації потрібних методів.

5.3 Реалізація модулю обчислення дискретного вейвлет-перетворення

Для реалізації функціоналу обчислення дискретного вейвлет-перетворення та виконання фільтрації даних на його основі був створений відповідний інтерфейс та його реалізація, а також декілька допоміжних перерахунків та клас-хелпер, що містить реалізацію материнських вейвлет-функцій, зображені на рисунку 5.3.

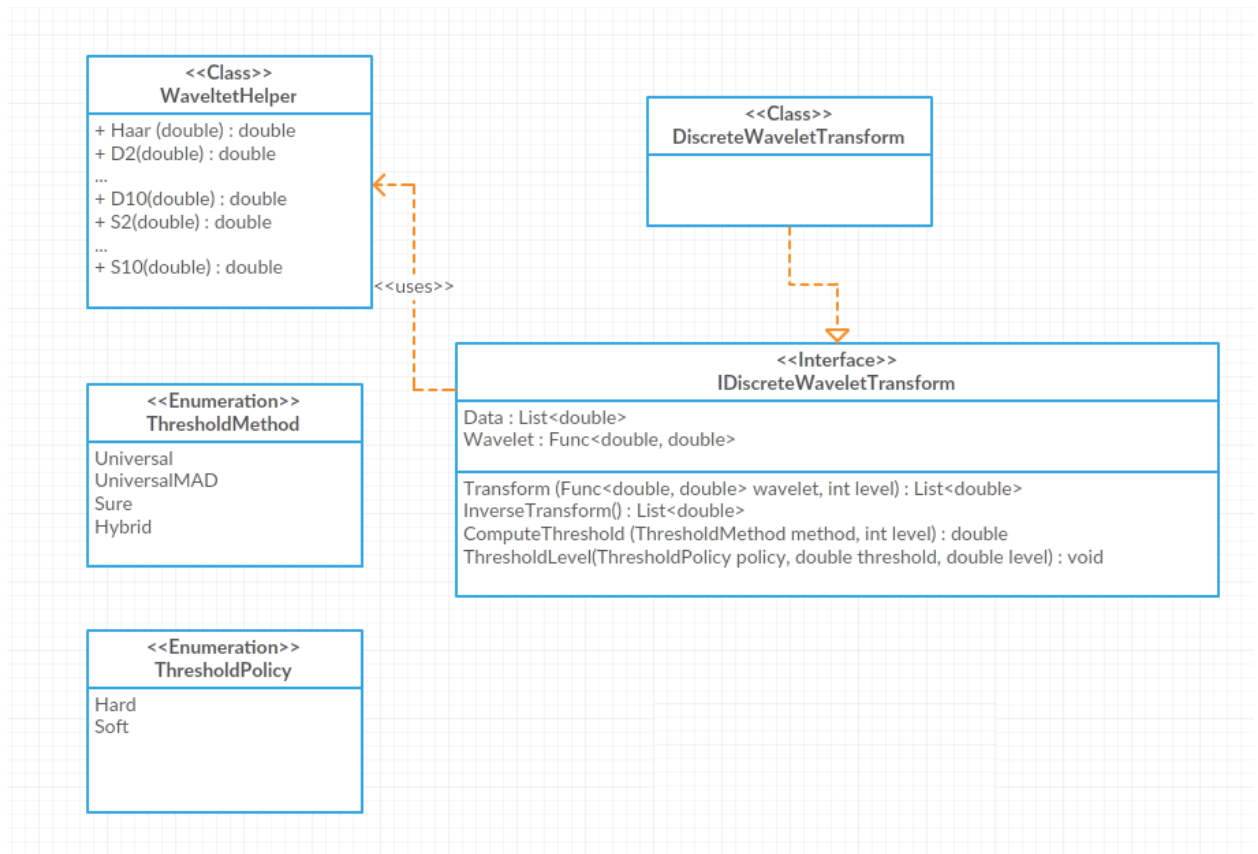


Рисунок 5.3 – Реалізація модулю обчислення дискретного-вейвлет перетворення

Розглянемо основні компоненти:

- **ThresholdMethod** містить список усіх можливих порогових методів, які система здатна застосувати.
- **ThresholdPolicy** містить список усіх можливих порогових політик, які система здатна застосувати.
- **WaveletHelper** – це статичний клас, який містить набір методів, які реалізують обчислення значень вейвлету.
- **Data** містить дані, які беруть участь у аналізі. Дане поле ініціалізується при створенні екземпляру класу, що реалізує інтерфейс.
- **Wavelet** містить функцію, що буде використовуватися, як материнська вейвлет-функція.
- **Transform** виконує дискретне вейвлет перетворення даних та розкладає їх до відповідного рівня.

- `InverseTransform` виконує зворотне дискретне вейвлет перетворення та відновлює сигнал за його розкладом.
- `ComputeThreshold` відповідає за обрахування порогового значення, беручи до уваги пороговий метод та рівень, для якого потрібно порахувати це значення.
- `ThresholdLevel` виконує фільтрацію певного рівня розкладу сигналу, беручи до уваги попередньо розраховане порогове значення та політику, яку слід застосувати при фільтрації.

Даний модуль повністю охоплює процес обчислення дискретного вейвлет перетворення та фільтрацію даних. В подальшому, за необхідності функціонал можна розширити шляхом наслідування та реалізації потрібних методів або додаванням до `WaveletHelper` нових функцій, які обраховують значення вейвлету.

5.4 Реалізація модулю обчислення швидкого перетворення Фур'є

Для реалізації функціоналу обчислення швидкого перетворення Фур'є був створений відповідний інтерфейс та його реалізація, зображені на рисунку 5.4.

Розглянемо його компоненти:

- `Transform` відповідає обчислення швидкого перетворення Фур'є та повертає `List<double>`, що містить частотну-часову характеристику обробленого сигналу.

Даний модуль повністю охоплює процес обчислення швидкого перетворення Фур'є. В подальшому, за необхідності функціонал можна розширити шляхом наслідування та реалізації потрібних методів.

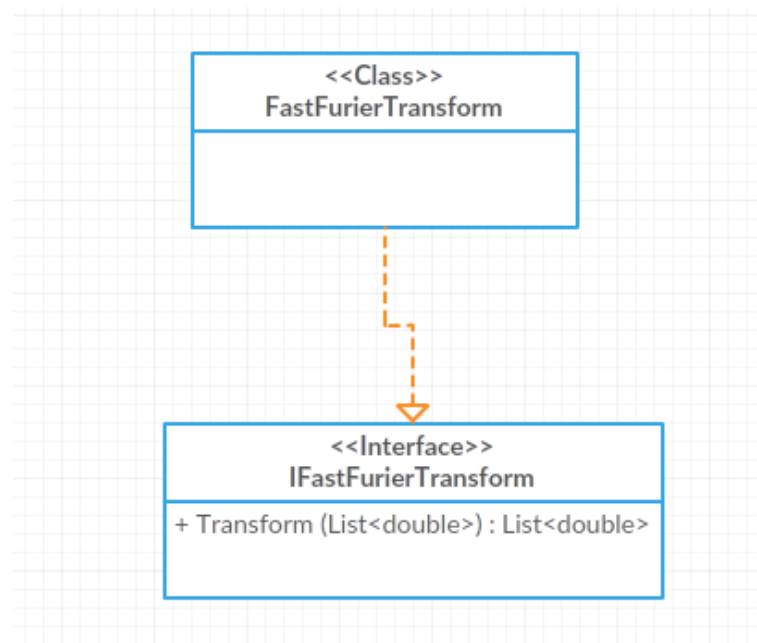


Рисунок 5.4 – Реалізація модулю обчислення швидкого перетворення Фур'є

5.5 Графічна інтерпретація даних

Для зручного відображення та полегшені аналізу отриманих результатів, вхідні та отримані в процесі обробки дані візуально відображаються за допомогою графіків. У випадку вхідного та відфільтрованого сигналів доречно відобразити дані у вигляді осцилограм, а для частотно-часового представлення обох сигналів зручно використати відображення у вигляді лінійного графіку, що містить інформацію про них одночасно. Для цього було використано бібліотеку для побудови графіків LiveCharts.

За допомогою елемента керування CartesianChart можна легко виконати побудову усіх вищезгаданих графіків. Для цього результати аналізів потрібно представити у вигляді LineSeries та додати отримані об'єкти до колекції, яка відображається на відповідному CartesianChart.

6 ПРИКЛАД РОБОТИ СИСТЕМИ

6.1 Інтерфейс програмного забезпечення

Візуальний інтерфейс розробленої системи (рисунок 6.1) складається з 5 секцій: панель завантаження сигналу, який потрібно відфільтрувати, панель конфігурацій, які будуть використовуватися при обробці сигналу, панель відображення осцилограми вхідного сигналу, панель відображення відфільтрованого сигналу та панель, на якій будується графіки дискретного перетворення Фур'є для обох сигналів.

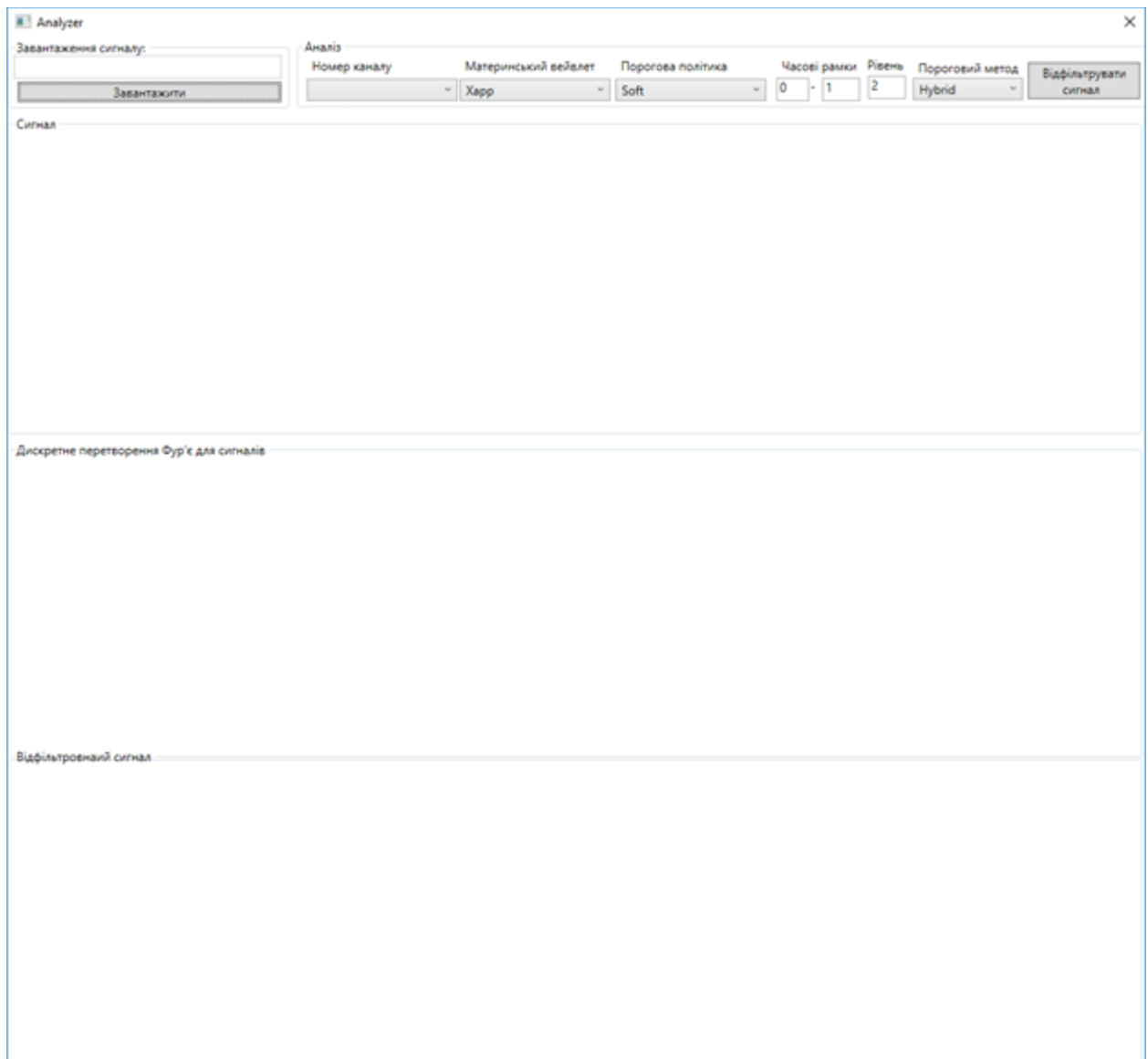


Рисунок 6.1 – Інтерфейс розробленого ПЗ

Для початку роботи користувачу потрібно завантажити сигнал, який потрібно обробити.

6.2 Завантаження сигналу до системи

Для того щоб завантажити сигнал до системи потрібно скористатися відповідною панеллю. Вона містить кнопку, яка відкриває діалогове вікно вибору сигналу у файловому сховищі комп'ютера (рисунк 6.2).

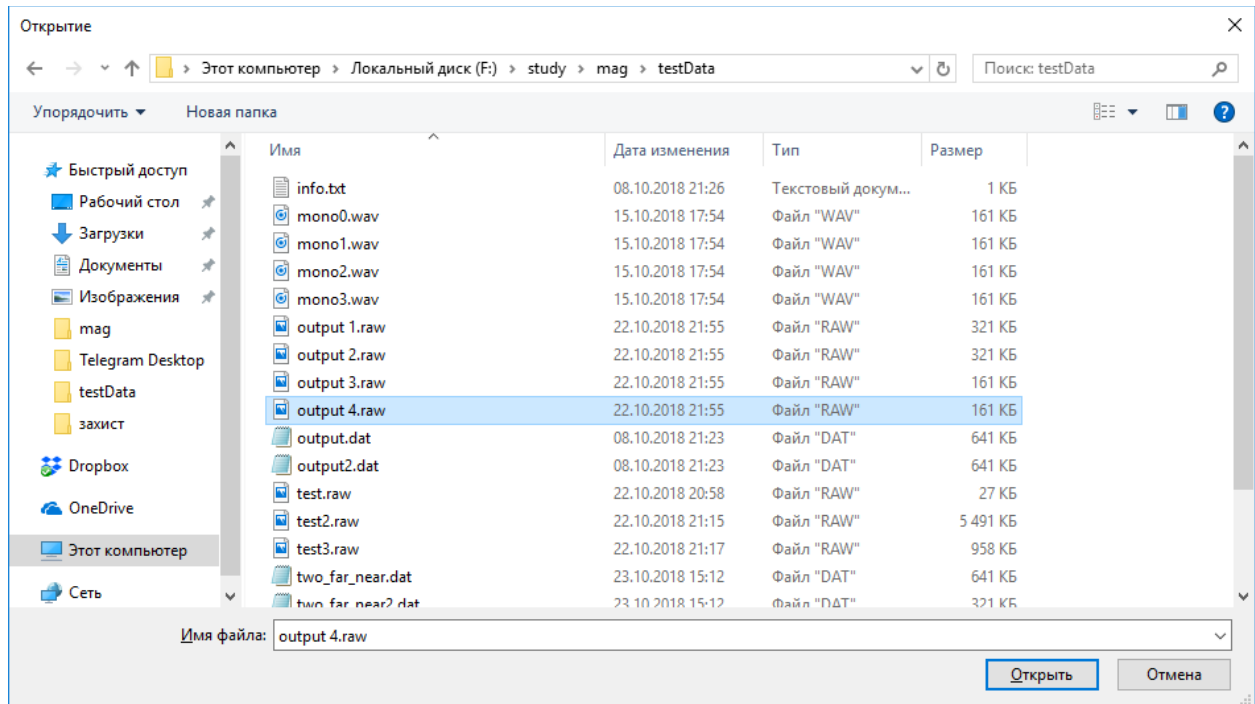


Рисунок 6.2 – Діалогове вікно вибору сигналу

У випадку, коли обраний файл не містить сигналу, який програма може обробити, програмне забезпечення відображає користувачу відповідне повідомлення та просить обрати коректний файл (рисунк 6.3).

Коли ж вміст файлу відповідає очікуванням, то дані обраного файлу завантажуються до оперативної пам'яті комп'ютера, а шлях до файлу відображається у відведеному для цього полі. Після цього можна переходити до вибору конфігурацій обробки завантаженого гідроакустичного сигналу.

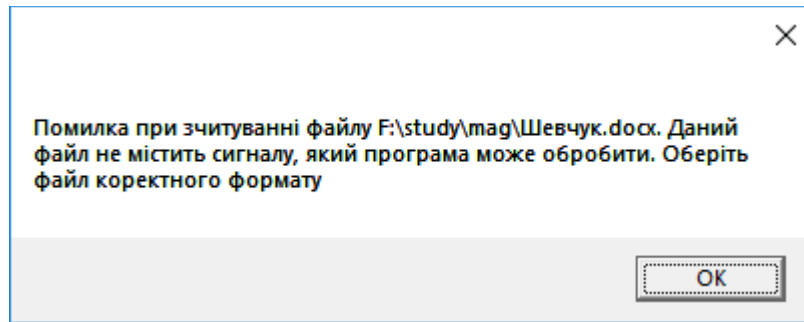


Рисунок 6.3 – Вікно, що сповіщає про некоректний вхідний файл

6.3 Конфігурування параметрів обробки сигналу

Після успішного імпорту сигналу до програмного забезпечення, користувачу варто звернутися до панель конфігурацій, яка знаходиться праворуч від панелі завантаження сигналу. В першу чергу, слід обрати номер каналу, який буде оброблятися. Після чого система автоматично побудує осцилограму відповідного каналу (рисунок 6.4).

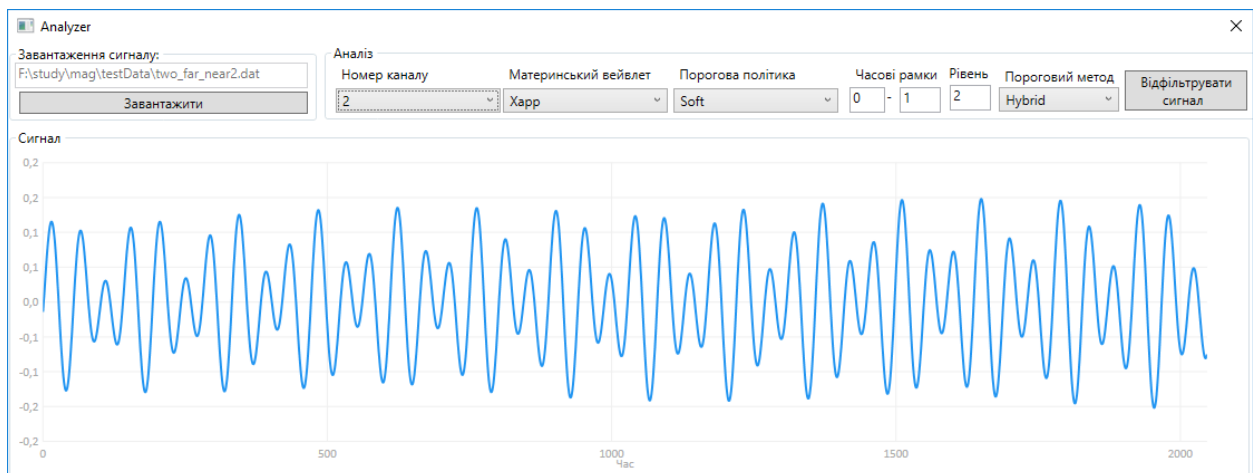


Рисунок 6.4 – Осцилограма другого каналу завантаженого файлу

Далі користувачу слід обрати материнський вейвлет на основі якого буде проводитися аналіз. Система передбачає наступні материнські вейвлети:

- Харра.
- Добеши 2-10 порядків.
- Симлет 2-10 порядків.

Наступним кроком буде вибір порогової політики, яку слід застосовувати при обробці. Користувачу запропоновано обирати її з випадającego списку, який містить:

- М'яку порогову політику.
- Жорстку порогову політику.

Для зручного вибору часового проміжку протягом, який буде оброблятися програмним забезпеченням, до панелі конфігурацій було додано поля для вводу початкової та кінцевої відміток часу. У ці комірки слід вводити дану у секундах. Наприклад, при введених 0 та 1, до полів початкової та кінцевої відміток відповідно, система буде брати до уваги сигнал від його початку до початку 2 секунди.

При спробі користувача ввести некоректні дані, система буде обробляти такі випадки та видавати відповідне повідомлення. У випадку коли користувач спробує ввести початковий відлік, який буде більше за кінцевий, то система сповістить про це та поверне попереднє значення до цього поля (рисунок 6.5). Наприклад, при кінцевій відмітці 2 секунди, користувач введе початкову зі значенням 4 секунди. Такий підхід працює і у зворотному напрямку.

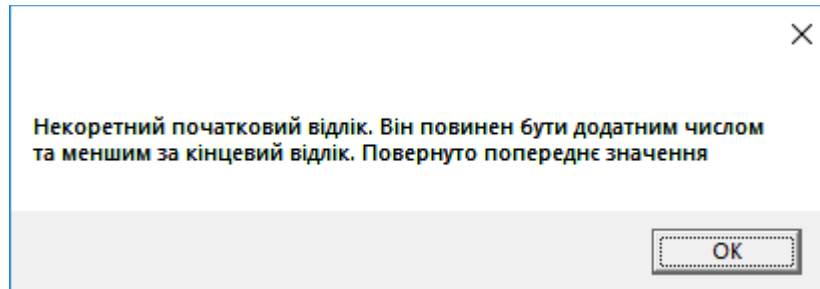


Рисунок 6.5 – Вікно, що сповіщає про некоректні часові рамки

При виборі часових рамок, слід зазначити, що сигнали, які завантажуються до системи мають дискретизацію у 2048 точок на секунду.

Важливою конфігурацією є кількість рівнів на які потрібно розкласти вхідний сигнал при дискретному вейвлет перетворені. Даний параметр обмежується материнським вейвлетом та об'ємом даних, які передаються на обробку. Для того щоб користувач не міг дати системі недопустиму кількість рівнів для поточного набору даних та обраних параметрів, система містить валідацію для цього поля. Коли полю буде встановлено некоректне значення, система поверне попереднє та сповістить про

це користувача. Наприклад, для завантаженого сигналу у якості материнського вейвлету було обрано вейвлет Хаара та аналіз проводиться на першій секунді сигналу, тобто на 2048 точках. При цьому користувач спробував розкласти сигнал на 20 рівнів. Для такого набору даних та параметрів максимально можливим рівнем є 11, тому система видасть сповіщення зображене на рисунку 6.6.

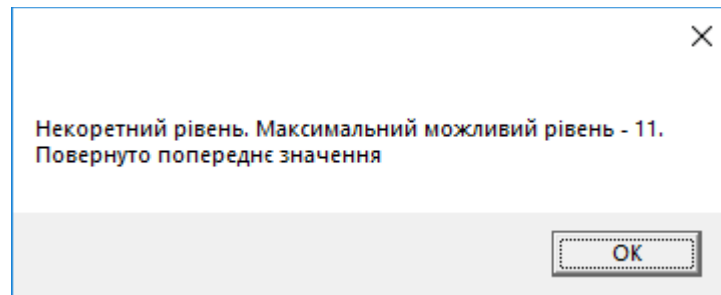


Рисунок 6.6 – Вікно, що сповіщає про некоректний рівень розкладу

Наступною конфігурацією є метод обчислення порогових значень. Система пропонує користувачу наступний перелік:

- Universal.
- UniversalMAD.
- Sure.
- Hybrid.

Коли було встановлено всі необхідні параметри, система готова до виконання фільтрації імпортованого сигналу. Для цього користувачу слід натиснути на кнопку «Відфільтрувати сигнал». Після чого програмне забезпечення проведе відповідні розрахунки та відобразить отримані результати у відповідних панелях вікна. В результаті обробки вхідного сигналу отримано відфільтрований сигнал, а його осцилограму відображено у нижній частині вікна та продемонстровано на рисунку 6.7.

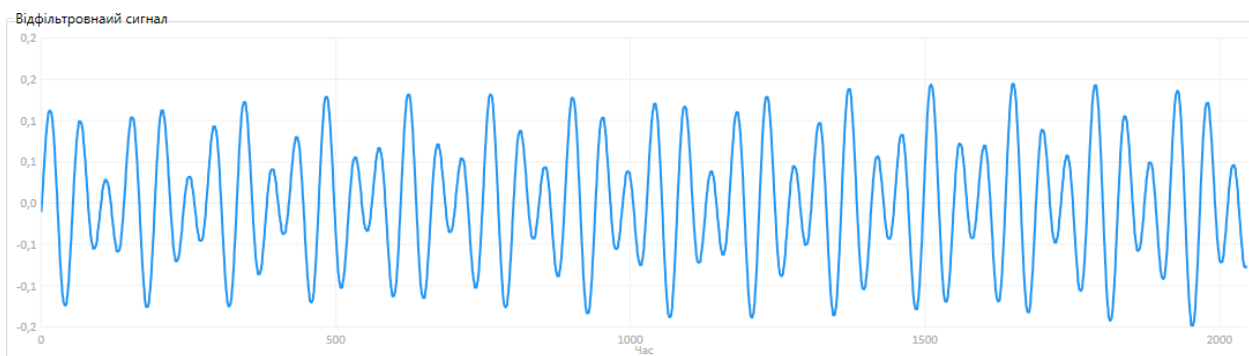


Рисунок 6.7 – Осцилограма відфільтрованого сигналу

Також при обробці сигналу, виконується перетворення Фур'є для вхідного та вже відфільтрованого сигналу, а результати відображено у середній частині вікна програми та продемонстровано на рисунку 6.8.

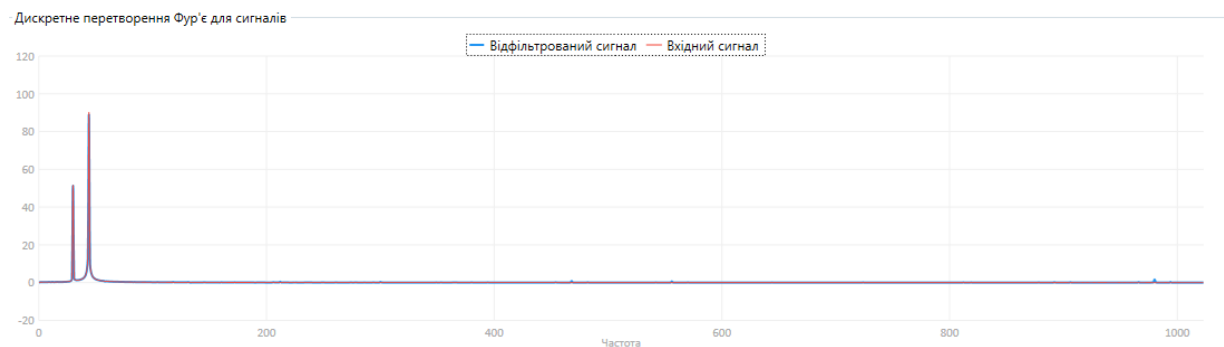


Рисунок 6.8 – Графік дискретного перетворення Фур'є для вхідного та відфільтрованого сигналу

При роботі з програмою слід пам'ятати, що для різних наборів даних слід обирати різні набори материнських вейвлетів та параметрів фільтрації, так як не існує універсального вибору для всіх сигналів. Для пришвидшення процесу аналізу даних було додано функцію автоматичного перерозрахунку результатів при зміні усіх параметрів, які беруть участь в процесі аналізу. А при зміні каналу, який потрібно відфільтрувати, результати попередніх дій зникають з робочої зони інтерфейсу, щоб не призвести до неправильного трактування отриманих результатів аналізу та фільтрації.

7 СТАРТАП

7.1 Опис ідеї проекту

Метою розділу є розроблення програмного забезпечення для аналізу гідроакустичних сигналів за допомогою вейвлетів. Зміст ідеї, можливі напрямки застосування, основні переваги, які зможе отримати користувач представлено у Таблиці 7.1.1.

Таблиця 7.1.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Програмне забезпечення для аналізу гідроакустичних сигналів за допомогою вейвлетів	Проведення вейвлет-аналізу	Дослідження гідроакустичних сигналів за допомогою вейвлет-аналізу
	Фільтрація гідроакустичних сигналів	Отримання незашумленого сигналу для подальшого їх аналізу
	Освітній процес	Використання даного продукту полегшить демонстрацію та ознайомлення студентів з методиками обробки гідроакустичних сигналів
	Наукові дослідження	Збір, збереження та аналіз даних, для дослідження гідроакустичних сигналів та формування кроків

Продовження табл. 7.1.1

		подальшого розвитку гідроакустики
--	--	--------------------------------------

Програмне забезпечення для аналізу гідроакустичних сигналів за допомогою вейвлетів призначена для аналізу, фільтрації та відображення отриманих результатів для даних гідроакустичних сигналів.

На ринку конкретні конкуренти відсутні, присутні тільки замітники у вигляді автоматизованих систем аналізу сигналів. Конкурент 1 - Wavelet Toolbox для MatLab, Конкурент 2 - Wavelet Explorer.

Таблиця 7.1.1 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко- економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		
		Мій проект	Конкурент 1	Конкурент 2
1	Технічні: швидкість роботи; точність результатів; можливість збереження результатів; режими доступу;	Висока швидкість аналізу роботи за рахунок проведення лише необхідних обчислень, які потрібні для поставленої задачі. Висока точність за рахунок ефективності обраних алгоритмів. Доступ без інтернету.	Середня швидкість аналізу, оскільки він проводиться паралельно з іншими обчисленнями для відображення результатів.	Середня швидкість аналізу, оскільки він проводиться паралельно з іншими обчисленнями для відображення результатів. Висока точність Збереження результатів в

Продовження табл. 7.1.2

			Висока точність. Збереження результатів в рамках однієї сесії. Доступ без інтернету.	рамках однієї сесії. Доступ через веб-сайт.
2	Економічі: ціна за підписку	Безкоштовна	Висока в межах галузі	Середня в межах галузі
3	Надійність	Не обмежений термін дії	Не обмежений термін дії	Не обмежений термін дії

Сильними сторонами системи відносно конкурентів є швидкість аналізу даних режим доступу та ціна за програмне забезпечення. До слабких сторін можна віднести відсутню можливість збереження результатів: у системі можливо лише проводити аналіз та переглядати результати. Нейтральними сторонами є необмежений термін дії.

7.2 Технологічний аудит ідеї проекту

В межах даного підрозділу проводиться аудит технологій, за допомогою яких можна реалізувати ідею проекту (технології створення товару).

Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (табл. 7.2.1):

Таблиця 7.2.1 – Технологічна здійсненність ідеї проекту

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Розробка програмного забезпечення для аналізу гідроакустичних сигналів за допомогою вейвлетів	Технологія 1. Самостійна розробка всіх компонентів програми без використання сторонніх бібліотек на мові C#.	Потребує розробки	Немає у відкритому доступі
	Технологія 2. Розробка компонентів програми з використанням функціоналу існуючих відкритих бібліотек (Live Charts, NMath).	Наявні	У відкритому доступі
	Технологія 3. Розробка компонентів програми з використанням функціоналу платформи MatLab для виконання необхідних	Наявні	Платні, з закритим кодом

Продовження табл. 7.2.1

	перетворень та їх відображення у зрозумілому для користувача форматі.		
--	---	--	--

Обрана технологія реалізації ідеї проекту: Технологія 2. Використання даної технології зменшить кількість розроблюваних блоків функціоналу та матеріальні затрати на реалізацію проекту.

7.3 Аналіз ринкових можливостей запуску стартап-проекту

Аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку представлено у таблиці 7.3.1

Таблиця 7.3.1 – Попередня характеристика потенційного ринку стартап-проекту.

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	5
2	Загальний обсяг продаж, грн/ум.од	10000 за рік
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Відсутні

Продовження табл. 7.3.1

5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
---	---	----------

Ринок є привабливий для входу та потребує новітніх засобів аналізу гідроакустичних сигналів.

Потенційні групи клієнтів, їх характеристика, та орієнтовний перелік вимог до товару для кожної групи представлений в таблиці 7.3.2.

Таблиця 7.3.2 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Аналіз гідроакустичних сигналів за допомогою вейвлетів	Науковці	Використання в процесі дослідження гідроакустичних сигналів	Швидкість роботи, точність результатів, отримання результатів потрібних для дослідження сигналів
	Студенти	Використання в процесі вивчення курсу гідроакустики	Зрозумілий і простий інтерфейс, точність результатів, низька ціна
	Викладачі	Викладання курсу гідроакустики	Зрозумілий і простий інтерфейс, точність результатів

Фактори, що сприяють ринковому впровадженню проекту, та фактори, що йому перешкоджають представлені в таблицях 7.3.3 та 7.4.4 відповідно.

Таблиця 7.3.3 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Збільшення ціни на розробку	Збільшення витрат на розробку, розширення технологічної бази.	Зміна цінової політики, додавання ексклюзивного функціоналу, перехід на більш перспективні та ефективні технології
2	Поява нових гравців на ринку	Поява на ринку конкурентів з аналогічними продуктами	Реклама, удосконалення продукту, додавання ексклюзивного функціоналу.

Таблиця 7.3.4 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Попит на програмний продукт	Потреба у якісному аналізі гідроакустичних сигналів сприяє збільшенню попиту на відповідне ПЗ.	Розробка та підтримка унікального нового надійного функціоналу
2	Зменшення ціни на розробку	Ефективніше використання наявних ресурсів.	Збільшення штату компанії, збільшення швидкості розробки.

Загальні риси конкуренції на ринку представлені в таблиці 7.3.5.

Таблиця 7.3.5 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції - олігополія	Домінує мала кількість компаній	Розробка унікального функціоналу, потужна рекламна кампанія.
2. За рівнем конкурентної боротьби: конкурентне середовище міжнародне	Боротьба ведеться на міжнародному ринку	Локалізація продукту, розвиток рекламної кампанії.
3. За галузевою ознакою - внутрішньогалузева	Продукт використовується у сфері аналізу гідроакустичних сигналів	Розробка унікального функціоналу, щоб повністю задовольнити користувачів
4. Конкуренція за видами товарів: - товарно-родова	Конкуренція між різними видами систем аналізу гідроакустичних сигналів.	Розробка унікального функціоналу.
5. За характером конкурентних переваг - не цінова	Ціна не є основним фактором конкуренції. Конкурентні переваги продукту засновані на його функціональних можливостях.	Впровадження нових функціональних можливостей

Продовження табл. 7.3.5

6. За інтенсивністю - не марочна	Продукт прив'язаний до відповідних розробок компанії (MatLab, Mathmatica)	Інтеграція нових можливостей та різних систем.
-------------------------------------	--	--

Аналіз конкуренції в галузі за М. Портером представлений в таблиці 7.3.6.

Таблиця 7.3.6 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари- замінники
Складові аналізу	Відсутні	Присутні	Відсутні	Споживачі диктують умови. Аналіз гідроакустичних сигналів є важливим аспектом розвитку гідроакустики	Присутні, зокрема Конкуренти 1,2.
Висновки	Прямі конкуренти на ринку відсутні.	Присутні. Вихід на ринок можливий.	Постачальники відсутні для розробки потрібні спеціалісти сфери розробки ПЗ.	Клієнти диктують умови. Якісний, дешевий, швидкодіючий продукт.	Обмежень для роботи на ринку товари- замінники не представляють

Робота на ринку можлива, відсутні прямі конкуренти, а ринок потребує системи даного типу.

На основі аналізу конкуренції, проведеного в таблиці 7.3.6, а також із урахуванням характеристик ідеї проекту (табл. 7.1.2), вимог споживачів до товару (таблиці 7.3.2) та факторів маркетингового середовища (таблиці №№ 7.3.3-7.3.4) визначено перелік факторів конкурентоспроможності. Представлених в таблиці 7.3.7.

Таблиця 2 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Швидкість аналізу	Аналіз проводиться за допомогою новітніх методів, що значно пришвидшує роботу.
2	Різносторонній аналіз	Система одночасно проводить різносторонній аналіз сигналу, що сприяє визначенню прихованих кореляцій.
3	Точність результатів	Висока точність результатів за рахунок правильного підходу до аналізу сигналі

За визначеними факторами конкурентоспроможності (табл. 7.3.7) проведений аналіз сильних та слабких сторін стартап-проекту представлений в таблиці 7.3.8., де К – конкурент, П – стартап-проект.

Таблиця 7.3.8 – Порівняльний аналіз сильних та слабких сторін проекту

№ п/п	Фактор конкурентоспроможності	Бал	Рейтинг товарів-конкурентів						
			-3	-2	-1	0	+1	+2	+3
1	Швидкість аналізу	20			К2	К1			П
2	Різносторонній аналіз	19				К2	К1	П	

Продовження табл. 7.3.8

3	Точність результатів	18				K1	K2	П	
4	Збереження результатів	17				П	K1	K2	
5	Доступність	15				K2	K1		П

SWOT-аналіз (матриця аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін приведено в таблиці 7.3.9.

Таблиця 7.3.93 – SWOT-аналіз стартап-проекту

Сильні сторони: швидкість аналізу, точність аналізу, можливість збереження результатів.	Слабкі сторони: доступність
Можливості: збільшення попиту на товар, зменшення коштів на розробку.	Загрози: збільшення ціни на розробку, нові гравці на ринку

На основі SWOT-аналізу розроблено альтернативи ринкової поведінки для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Результати приведені в таблиці 7.3.10.

Таблиця 4.10 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Розробка базової версії продукту, з забезпеченням	висока	До 2 місяців

Продовження табл. 7.3.10

	запланованого функціоналу.		
2	Розробка версії продукту з використанням власної системи аналізу даних.	низька	До 6 місяців
3	Створення онлайн версії продукту	середня	До 3 місяців
4	Впровадження найновіших ефективних технологій	низька	До 5 місяців

Обрана альтернатива під номером один, оскільки її реалізація є найбільш вірогідна та з малими строками.

7.4 Розробка ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку. Опис цільових груп потенційних споживачів представлений в таблиці 7.4.1.

Таблиця 5.1 – Вибір цільових груп потенційних споживачів

Опис профілю цільової групи потенційних клієнтів.	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
---	---	---	--------------------------------------	--------------------------

Продовження табл. 7.4.1

Інститути та університети, що займаються аналізом гідроакустичних даних	Споживачі готові сприйняти продукт.	Попит високий в цільовому сегменті	Конкуренція з товарами-замінниками	Продукт простий у впровадження його на ринок
---	-------------------------------------	------------------------------------	------------------------------------	--

Обрана стратегія концентрованого маркетингу, оскільки присутній один цільовий сегмент.

Визначення базової стратегії розвитку представлено в таблиці 7.4.2

Таблиця 7.4.26 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Розробка базової версії функціоналу, з подальшим покращенням.	Стратегія розширення первинного попиту	Конкурентами є лише товари-замінники. Розроблювана система має переваги в швидкості, точності та простоті використання.	Стратегія диференціації

Вибір стратегії конкурентної поведінки представлений в таблиці 7.4.3.

Таблиця 7.4.37. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопроходцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
Ні	Шукати нових та забирати споживачів у конкурентів	Ні	Стратегія лідера

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (таблиця 7.3.2), а також в залежності від обраної базової стратегії розвитку (таблиця 7.4.2) та стратегії конкурентної поведінки (таблиця 7.4.3) розроблена стратегія позиціонування представлена в таблиці 7.4.4.

Таблиця 7.4.4 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту (три ключових)
	Швидкість, точність, унікальний функціонал	Стратегія диференціації	Швидкість, точність, простота використання	Швидкість, точність, сучасність

7.5 Розроблення маркетингової програми стартап-проекту

Таблиця 7.5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Швидкодія	Швидкодія за рахунок використання новітніх алгоритмів аналізу сигналів.	Процес аналізу швидший ніж у конкурентів.
2	Точність	Точність за рахунок великої новітніх алгоритмів аналізу сигналів.	Диференціація експериментів за певними показниками та визначення найкращих по певному показнику.
3	Простота використання	Інтуїтивно зрозумілий неперевантажений інтерфейс	Конкурентні рішення мають доволі перевантажений та складний інтерфейс.
4	Різносторонній аналіз	Система дозволяє знайти приховані кореляції.	Конкуренти працюють тільки з одним типом показників одночасно.

Трирівнева маркетингова модель товару представлена в таблиці 7.5.2.

Таблиця 7.5.2 9– Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Програмне забезпечення для аналізу гідроакустичних сигналів за допомогою вейвлетів.

Продовження табл. 7.5.2

	<p>Програмне забезпечення для аналізу гідроакустичних сигналів за допомогою вейвлетів.</p> <p>Можна виділити наступні вигоди від використання продукту:</p> <ul style="list-style-type: none"> • Швидкість аналізу даних; • Різносторонній аналіз; • Точність; • Простота використання; 		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Висока швидкодія	М	Тх/Тл
	2. Висока точність	М	Тх/Тл
	3. Різносторонній аналіз	М	Тх/Тл
	4. Простота використання	М	Вр/Тл
	Якість: відповідає нормам розробки програмного забезпечення.		
	<p>Пакування: Продукт представлений у десктопного додатку.</p> <p>На сайті міститься:</p> <ul style="list-style-type: none"> • загальна назва продукту, власна назва; • опис продукту; • функції, які представлені; <p>інструкція для користування;</p> <ul style="list-style-type: none"> • контакти для зв'язку з розробником; • підтримка для клієнтів. 		
	Марка: ПП «HS analyzer»		

Проект буде захищений як інтелектуальна власність.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар. Результат представлений в таблиці 7.5.3.

Таблиця 7.5.310 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар
15 тис. грн	Відсутні	600 тис. грн	5 тис. грн – 7 тис. грн

Визначення оптимальної системи збуту представлено в таблиці 7.5.4.

Таблиця 7.5.4 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Реєстрація на сайті, оплата продукту, відсилання на електрону пошту ключа активації.	Доставка на електрону пошту ключа активації.	Виробник - споживач	Web-сайт

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів. Результат представлений в таблиці 7.5.5.

Таблиця 7.5.5 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій цільових клієнтів	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
---------------------------------------	--------------------------------------	--	----------------------------------	--------------------------------

Продовження табл. 7.5.5

Ретельно обирають систему	Веб-сайт, телефон, месенджер	Підтримка, індивідуальний підхід, постійне оновлення	Донести переваги продукту до клієнта	Вейвлет-аналіз гідроакустичних сигналів.
---------------------------------	------------------------------------	---	---	--

ВИСНОВКИ

Вейвлет-перетворення надає найбільш наочну та інформативну картину результатів експерименту, дозволяє очистити вихідні дані від шумів і випадкових спотворень, і навіть наочно помітити деякі особливості даних і напрямки їхньої подальшої обробки та аналізу. Крім того, вейвлети добре підходять для аналізу складних сигналів.

Завдяки високій ефективності алгоритмів та стійкості до впливу перешкод, вейвлет-перетворення є потужним інструментом у тих областях, де традиційно використовувалися інші методи аналізу даних, наприклад, перетворення Фур'є. На відміну від традиційних методів, вейвлет-аналіз дає інформацію не лише про присутність певної частоти, а й в який проміжок часу ця частота присутня. Можливість застосування вже існуючих методів обробки результатів перетворення, а також характерні риси поводження вейвлет-перетворення в частотно-часовій області дозволяють істотно розширити й доповнити можливості подібних систем.

У результаті виконаної роботи розроблено десктопний застосунок, який дозволить користувачам відфільтрувати завантажений гідроакустичний сигнал та за допомогою можливості конфігурування параметрів фільтрації підібрати найбільш влучний набір для її виконання.

Систему розроблено на мові програмування C# з використанням технології WPF. Для побудови графіків було використано бібліотеку LiveCharts, а для полегшення математичних розрахунків бібліотеку NMath. Реалізована архітектура системи забезпечує легку підтримку, заміну та модернізацію її компонентів, що дозволяє ввести нові функції та розширити вже існуючі.

Результати роботи презентувались на XVI міжнародній науково-практичній конференції аспірантів, магістрантів, студентів «Сучасні проблеми наукового забезпечення енергетики» [30] та пройшли апробацію.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Непрерывное wavelet преобразование / Хабрахабр [Электронный ресурс] – Режим доступа: <https://habrahabr.ru/post/103899/>
2. Латфуллин И. А. Вейвлет анализ частотных характеристик ППЖ по грудным отведениям ЭКГ / И. . Латфуллин, Г. М. Тептин, Л. Э. Мамедова // Тезисы Всероссийского конгресса "Неинвазивная электрокардиология в клинической медицине", Москва, 19-20 квітня 2007 р.
3. Filtering Out The Noise | Signal Processing Хабрахабр [Электронный ресурс] – Режим доступа: <http://blog.catchpoint.com/2016/06/29/signal-vs-noise/>
4. K. P. Obulesu P.U. kumar, "Implementation of time frequency block thresholding algorithm in audio noise reduction," international journal science, engineering and technology research, pp.1513-1520, july 2013.
5. J. Jebastine, B. S. Rani, "Design and implementation of noise free Audio speech signal using fast block least Mean square algorithm", Signal & Image Processing: An International Journal (SIPIJ), Vol. No.3, pp.39-53, june 2012.
6. S. Chakrabarty, S. Maitra, "Design of IIR Digital High pass Butterworth Filter using Analog to Digital Mapping Technique", International Journal of Computer Applications ISSN: 0975 –8887, Volume 52 –No. 7, pp.19-26, November 2012
7. Wavelet Toolbox – MatLab [Электронный ресурс] – Режим доступа: <https://www.mathworks.com/products/wavelet.html>
8. Mathcad Wavelets Extension Pack – Mathcad Add-On [Электронный ресурс] – Режим доступа: <http://www.adeptscience.co.uk/products/mathsim/mathcad/mathcad-wavelets-extension-pack.html>
9. Вейвлет-анализ: Новое в системе Wolfram Mathematica 8 [Электронный ресурс] – Режим доступа: <https://www.wolfram.com/mathematica/new-in-8/wavelet-analysis/>
10. Афонский А. А., Дьяконов В. П. Цифровые анализаторы спектра, сигналов и логики / Под ред. проф. В. П. Дьяконова. — М.: СОЛОН-Пресс, 2009. — С. 248
11. Сергиенко А. Б. Цифровая обработка сигналов. — 2-е изд. — СПб.: Питер, 2006. — С. 751

12. An introduction to wavelet transforms: a tutorial approach [Электронный ресурс] – Режим доступа: <https://pdfs.semanticscholar.org/8960/a36e956b996d7f1639ff577d4b21f389ded9.pdf>
13. Малла С. Вейвлеты в обработке сигналов. М.: Мир, 2005.
14. Астафьева Н.М. Вейвлет-анализ: основы теории и примеры применения //УФН. - 1996. - Т.166. - № 11. - С. 1145-1170.
15. Анализ_сигналов_на_основе_вейвлет-преобразования – Национальная библиотека им. Н. Э. Баумана [Электронный ресурс] – Режим доступа: https://ru.bmstu.wiki/Анализ_сигналов_на_основе_вейвлет-преобразования
16. Вейвлеты. От теории к практике. Изд. 2-е, перераб. и доп.— М.: СОЛОН-Пресс, 2010. — 400 с.: ил. — (Серия «Полное руководство пользователя»).
17. Добеши И. Десять лекций по вейвлетам. – Ижевск: НИЦ Регулярная и хаотическая динамика, 2001. – 464 с.
18. Zhao Q, Dai W. A wavelet denoising method of new adjustable threshold. 2015 IEEE 16th International Conference on Communication Technology (ICCT); 2015; pp. 684–688.
19. Johnstone I, Silverman B. Wavelet threshold estimators for data with correlated noise. J Roy Statist Soc D. 1997;59:319–351.
20. Chang F, Hong W, Zhang T, Jing J, Liu X. Research on wavelet denoising for pulse signal based on improved wavelet thresholding. Pervasive Computing Signal Processing and Applications (PCSPA), 2010 First International Conference on; Sept 2010; pp. 564–567.
21. Cohen R. Technion, Israel Institute of Technology, Tech Rep. 2012. Signal denoising using wavelets.
22. Basic Line Chart [Электронный ресурс] – Режим доступа: <https://lvcharts.net/App/examples/v1/wpf/Basic%20Line%20Chart>
23. NMath User's Guide [Электронный ресурс] – Режим доступа: <https://www.center-space.net/doc/NMath/user/>

24. Уотсон, Карл и, Нейгел, Кристиан, Педерсен, Якоб Хаммер, Рид, Джон Д., Скиннер, Морган, Уайт, Эрик. Visual C# 2008: базовый курс. : Пер. с англ. - М. : ООО "И.Д. Вильяме", 2009. - 1216 с.: ил. — Парал. тит. англ.
25. WPF - Overview [Электронный ресурс] – Режим доступа до ресурсу: https://www.tutorialspoint.com/wpf/wpf_overview.htm
26. Справочник «Паттерны проектирования» [Электронный ресурс] – Режим доступа до ресурсу: <http://design-pattern.ru/patterns/mvvm.html>
27. Обзор XAML (WPF) [Электронный ресурс] – Режим доступа до ресурсу: <https://docs.microsoft.com/ru-ru/dotnet/framework/wpf/advanced/xaml-overview-wpf>
28. Inversion of Control Containers and the Dependency Injection pattern [Электронный ресурс] – Режим доступа до ресурсу: <http://martinfowler.com/articles/injection.html>.
29. Visual Studio IDE [Электронный ресурс] – Режим доступа: <https://visualstudio.microsoft.com/ru/>
30. Матеріали XVI Міжнародної науково-практичної конференції аспірантів, магістрантів і студентів, м. Київ, 25-28 квітня 2018 року [Електронний ресурс] – Режим доступу до ресурсу: http://tef.kpi.ua/files/pdf/1tezi_tom2_1524728034.pdf.

ДОДАТОК А

Апробація на конференції «Сучасні проблеми наукового забезпечення енергетики»

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XVI Міжнародної
науково-практичної конференції
аспірантів, магістрантів і студентів
м. Київ, 24-27 квітня 2018 року,

ТОМ 2



Київ- 2018

<i>ГВОЗДЕНКО О.В., магістрант гр. ПІ-71мп</i> <i>Керівник - доц., к.т.н. Стативка Ю.І.</i>	
Методи та засоби аналізу надвеликих масивів даних у реальному часі.	239
<i>ЛИСТОПАДОВ Д.О., магістрант гр. ТМ-71мп</i> <i>Керівник - доц., к.т.н. Коваль О.В.</i>	
Формування сценаріїв розпізнавання інформаційних вкидань на основі нейронних мереж.	240
<i>МАГЕРАМОВА І.І., магістрант гр. ТР-71мп</i> <i>Керівник - доц., к.ф.-м.н. Коваль О.В.</i>	
Інструментальні засоби фінансового аналізу бюджету.	241
<i>МАРЧИШИНА О.В., магістрант гр. ТР-71мп</i> <i>Керівник - доц., к.т.н. Коваль О.В.</i>	
Відновлення траєкторії руху морських об'єктів векторно-фазовим методом.	242
<i>МОРОЗОВ М.С., магістрант гр. ПІ-71мп</i> <i>Керівник - доц., к.т.н. Варава І.А.</i>	
Синтаксичний аналіз простих речень.	243
<i>МУЗИКА В.В., магістрант гр. ПІ-71мп</i> <i>Керівник - доц., к.т.н. Стативка Ю.І.</i>	
Методичні основи розробки системи моделювання ризиків досягнення сталого розвитку України .	244
<i>ФЕДОРЕНКО В.Є., магістрант гр. ТМ-71мп</i> <i>Керівник - доц., к.в.н. Карасєва Н.В.</i>	
Характеристика методів аналізу ризиків в енергетиці.	245
<i>ФЕДЧИШИН М.О. магістрант гр. ТМ-71мп</i> <i>Керівник - доц., к.в.н. Карасєва Н.В.</i>	
Реінжиніринг бізнес-процесів діяльності кафедри.	246
<i>ФЕДЬКІН С.С., магістрант гр. ТР-71мп</i> <i>Керівник - доц., к.т.н. Кузьмінєх В.О.</i>	
Обробка гідроакустичних сигналів за допомогою вейвлетів.	247
<i>ШЕВЧУК О.О., магістрант гр. ПІ-71мп</i> <i>Керівник - доц., к.т.н. Варава І.А.</i>	
Автоматична класифікація текстів за жанровими ознаками.	248
<i>ІЛЬЧИШИН Д.В., магістрант гр. ТВ-71мп</i> <i>Керівник - доц., к.т.н. Стативка Ю.І.</i>	
Система аналізу ризиків радіаційної безпеки.	249
<i>ВЛАСЕНКО Н.А., студент гр. ТМ-41</i> <i>Керівник - ст. викл. Бандурка О.І.</i>	
Система для аналізу і оптимізації поширення госпітальних інфекцій в лікарняному відділенні.	250
<i>ВІННИЧУК М.В., студент гр. ТВ-41</i> <i>Керівник - ст. викл. Бандурка О.І.</i>	
Програма для графічного порівняння витрат і корисності діагностичних методів .	251
<i>ВІННИЧУК М.В., студент гр. ТВ-41</i> <i>Керівник - ст. викл. Бандурка О.І.</i>	
Середовище для написання SQL запитів.	252
<i>ГОЛЬДИЧ Я.Є., студент гр. ТВ-42</i> <i>Керівник - доц., к.т.н. Стативка Ю.І.</i>	
Енергоаудит системи опалення багатоквартирного будинку - методика, ефективність результатів.	253
<i>ГУРСЄВ М.В., студент гр. ОТ-41</i>	

УДК 621.43.056:632.15

Магістрант 5 курсу, гр. ПІ-71мп Шевчук О.О.
Доц., к.т.н. Варава І.А.

ОБРОБКА ГІДРОАКУСТИЧНИХ СИГНАЛІВ ЗА ДОПОМОГОЮ ВЕЙВЛЕТІВ

Гідроакустика отримала широке практичне застосування – її використовують для підводної локації, зв'язку, океанологічних досліджень, вирішення військових задач. Ніякі види електромагнітних хвиль не поширюються у воді на значні відстані, тому звук є єдиним можливим засобом зв'язку під водою.

Одним із способів вирішення поставлених перед гідроакустикою задач є використання вейвлетів. Вейвлет-перетворення було створено як інструмент, який вирішує проблему невизначеності Гейзенберга для побудови частотно-часових характеристик сигналу. Його альтернативою є перетворення Фур'є. Але воно дає інформацію тільки про частоту, яка присутня в сигналі, і не дає ніякої інформації про те, в який проміжок часу ця частота присутня. Таким чином перетворення Фур'є за своєю суттю не може відрізнити стаціонарний сигнал від нестаціонарного, що є великою проблемою для його застосовності [1].

Саме тому система, що розробляється, базується на вейвлет-перетворенні. Вона пропонує користувачу завантажити аудіофайл гідроакустичного сигналу в форматі wav та виконати його вейвлет-аналіз. Після обробки завантажених даних ми отримуємо візуальне відображення у вигляді скейлограми. Вона складається з трьох осей для представлення часу (відліків), масштабу та значень вейвлет-коефіцієнтів [2]. На основі скейлограми можна визначити зміни частоти в часі, які свідчать про зміну швидкості морського об'єкта. Для обробки сигналу використовується вейвлет-перетворення у вигляді:

$$W(a, b) = \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt, \quad (1)$$

де $f(t)$ – дані про сигнал, що аналізуються і залежать від часу t , ψ – вейвлет; a і b – параметри масштабу і часу для вейвлет-перетворення; $a, b \in \mathbb{R}$; $a > 0$.

В результаті перетворення отримано матрицю вейвлет-коефіцієнтів $W\langle i, b_k \rangle$, яка відображає ступінь присутності певної частоти в конкретний момент часу, де i відповідає номеру масштабу, а k – номер часового відліку для вейвлет-перетворення. На основі цих даних можна побудувати скейлограми $V(a_i)$:

$$V_i\langle a_i \rangle = \frac{1}{N} \sum_{k=1}^N W^2\langle i, b_k \rangle \quad (2)$$

де N – число усереднених вейвлет-коефіцієнтів $W\langle i, b_k \rangle$.

Відмінною особливістю вейвлет-аналізу є те, що в ньому можна використовувати сімейства функцій, що реалізують різні варіанти співвідношення невизначеності. Відповідно, користувач має можливість гнучкого вибору між ними і застосування тих вейвлетних функцій, які найбільш ефективно вирішують поставлені завдання. Таким чином, враховуючи багатоканальність гідроакустичного сигналу та результати, які можна отримати після їх вейвлет-аналізу, можна отримати значних успіхів у вирішенні задач поставлених перед гідроакустикою – визначення напрямоку руху і відстані до об'єкту.

Перелік посилань:

1. Непрерывное wavelet преобразование / Хабрахабр [Електронний ресурс] – Режим доступу: <https://habrahabr.ru/post/103899/>
2. Scaelogram – Wikipedia [Електронний ресурс] – Режим доступу: <https://en.wikipedia.org/wiki/Scaleogram>

ДОДАТОК А

Публікації

Обробка гідроакустичних сигналів за допомогою вейвлетів

УКР.НТУУ"КП" _ТЕФ_АПЕПС_ ТІЗ190_18М

Аркушів 3

2018

ДОДАТОК Б

Акт впровадження

Обробка гідроакустичних сигналів за допомогою вейвлетів

УКР.НТУУ"КПІ"_ТЕФ_АПЕПС_ ТІЗ190_18М

Аркушів 2

2018